Induttanzimetro MICROCONTROLLORI Visione artificiale Display LCD USB PLC PIC-based Progammazione PIC Pascal Interrupt PIC18 CONTATORE JOHNSON Raspberry PI VXO WEB SERVER
fe
fare elettronica
www.farelettronica.com
332-333 Febbraio/Marzo Mensile Anno XXIX € 6,00
web server con
ARDUINO UNO
SUPERVXO
MODIFICATO
TECNICHE
DI MONTAGGIO
SU MILLEFORI
Tutorial PIC18
gestione degli interrupt
FILTRI DIGITALI
UTILIZZARE GLI I/O
DI RASPBERRY PI
INDUTTANZIMETRO
DIGITALE
ISSN 1591-2272
30333
9 771591 227008

Addio 7805 e 7905

I modelli switching equivalenti sono migliori

I regolatori lineari di tensione, di tutte le forme e dimensioni, sono utilizzati in un numero enorme di circuiti elettronici. Dispositivi classici come il 7805 con uscita positiva e il 7905 con uscita negativa sono indispensabili nella maggior parte dei circuiti, ma essi non sono molto efficienti dal punto di vista energetico a causa della loro modalità di funzionamento lineare. Tutta la potenza superflua tra i pin di ingresso e di uscita di questi regolatori viene infatti convertita in calore. Sicuramente oggi possiamo fare di meglio: maggiore efficienza energetica e maggiore rispetto per l'ambiente? Questo è proprio l'argomento di questo articolo. Il messaggio è chiaro: abbasso la dissipazione di potenza, evviva lo switching!

sommario

332/333 *febbraio/marzo 2013*

Rispondi e... VINCI! *pag.* 48

DIVERTITI E METTI ALLA PROVA LE TUE CONOSCENZE CON **ELETTRO QUIZ** E VINCI OGNI MESE ESCLUSIVI PREMI!

di MAURIZIO DEL CORSO

Al passo con i tempi

Vista la velocità esponenziale con cui si evolve la tecnologia, stare al passo non è semplice ma è un dovere soprattutto per una rivista che parla di tecnologia.

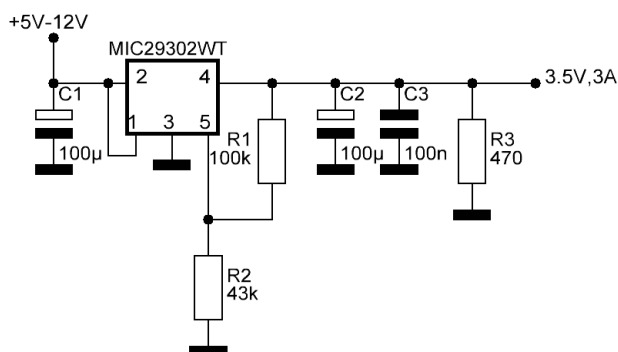
Immaginatevi lo spot pubblicitario di uno smartphone in cui una persona effettua una telefonata usando un vecchio telefono fisso a filo: un controsenso, no? Bene questo è quello che sta succedendo nel campo dell'editoria tecnica: si parla di nuove tecnologie, di applicazioni che consentono di controllare dispositivi tramite i tablet o gli smartphone, si insegna a costruire un web server, ma tutto questo usando un mezzo di comunicazione veramente obsoleto come la carta. Si ricorre quindi ad escamotage come i QRcode per integrare gli articoli con le risorse multimediali online, ovviamente impossibili da riportare sulla carta in altro modo. Da qui la scelta che noi di Fare Elettronica abbiamo adottato: passare alla versione digitale. Solo in questo modo riusciremo nei prossimi numeri a darvi non solo i contenuti interessanti di sempre, ma anche un mondo di altre informazioni altrimenti impossibili da inserire sulla carta. Crediamo che i tempi siano maturi, che i PC, gli smartphone e i tablet siano sufficientemente diffusi, che i nostri lettori siano molto "smart" e già pronti al cambiamento. Crediamo soprattutto che una rivista che parla di tecnologie, deve essere la prima ad utilizzarle. Il prossimo numero di Fare Elettronica sarà dunque disponibile solo in formato pdf, sarà l'inizio di una nuova era in cui gli abbonati non si lamenteranno più dei ritardi delle poste, in cui il nostro impegno sarà ai massimi livelli per sfruttare al meglio tutte le nuove possibilità; l'era in cui la vostra rivista preferita non andrà mai persa perché entrerà a far parte del gigantesco cloud di contenuti sempre disponibili online, in qualsiasi luogo ed in qualsiasi momento.
Adesso viene il bello...

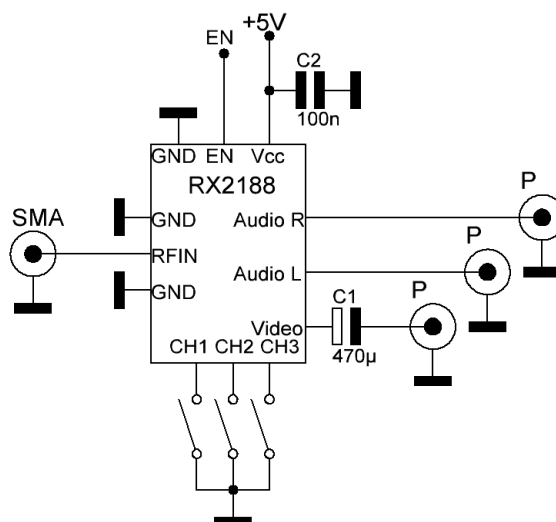
di FRANCESCO DI LORENZO

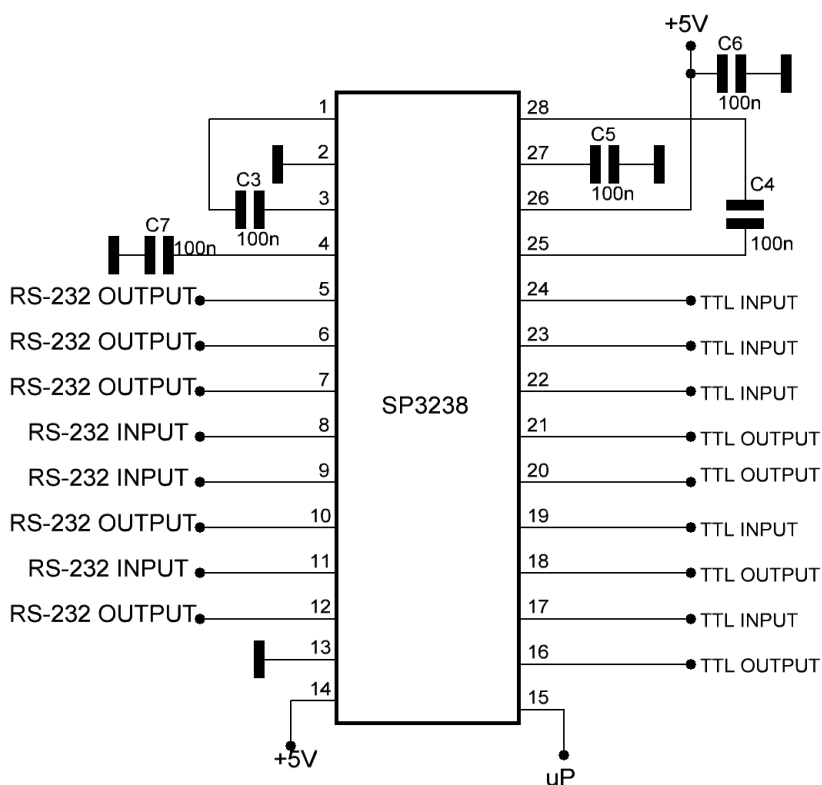
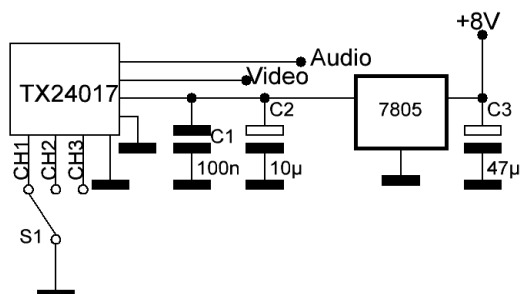
ALIMENTATORE STEP-DOWN

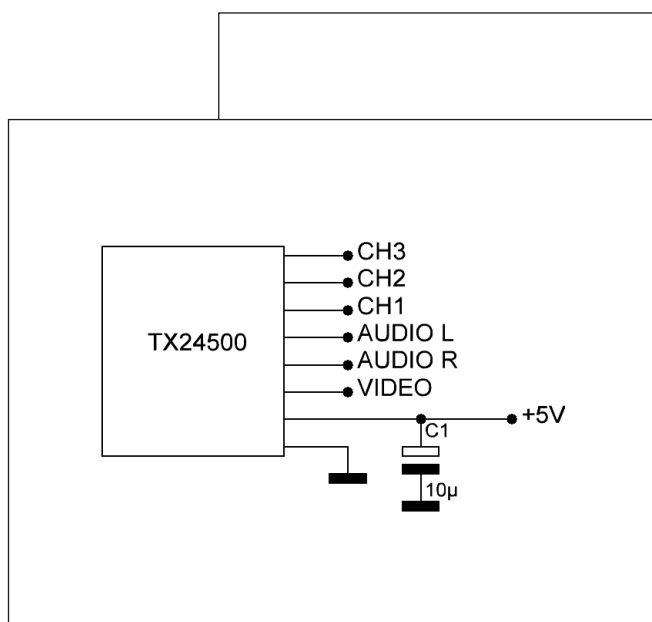
Nella figura è riportato lo schema di un alimentatore step-down realizzato con un regolatore della ST, siglato LM2576. Questo integrato è largamente utilizzato per il semplice fatto che in ingresso è possibile applicare una tensione che varia dai +7V ai +40V ed in uscita è possibile ottenere 5V, con una corrente massima erogabile è di 3A.

REGOALTORE STEP-DOWN REALIZZATO CON MIC29302WT

Il circuito in figura è un stabilizzatore step- down che accetta in ingresso tensioni non superiori ai +7V e fornisce in uscita una tensione variabile con una corrente massima di 3A. Viene realizzato con forme e dimensioni differenti a seconda delle varie necessità dei progettisti. Trova largo impiego nel campo della telefonia, nei carica batteria.

RICEVITORE WI-FI

Il ricevitore in figura è realizzato con un RX2188, che è un modulo radio di dimensioni ridotte, utilizzato nel campo della trasmissione di dati audio e video. Opera con una frequenza che varia dai 2400MHz ai 2480MHz. Questo range di frequenza è suddiviso in 4 bande ognuna corrispondente a un canale specifico , in questo caso si hanno 4 canali selezionabili mediante DIP-SWITCH. La tensione di funzionamento non deve superare i 5V.

ADATTATORE DI SEGNALE TTL-CMOS PER RS-232

L'SP3238 è un dispositivo destinato all'adattamento dei segnali TTL o CMOS nello standard RS232. Questo circuito è in grado di trasmettere fino a 5 segnali contemporaneamente e di riceverne altri 5 contemporaneamente. Deve essere alimentato con una tensione non superiore ai 6V.

TRASMETTITORE AUDIO VIDEO WI-FI 4 CANALI

Il trasmettitore raffigurato è stato realizzato con un TX24017. Questo trasmettitore opera con un range di frequenze 2414-2468MHz, suddiviso in quattro bande ciascuna delle quali forma un canale, selezionabile mediante DIP-SWITCH. Questo trasmettitore viene utilizzato nella trasmissione di segnali audio e video, per applicazioni di videosorveglianza o di audio diffusione. La sua potenza massima è di circa 50 mW, necessaria a ricoprire una distanza di quasi 100 metri.

TRASMETTITORE WI-FI A 8 CANALI

Nella circuito seguente viene riportato un trasmettitore audio-video a 8 canali con una potenza di 500mW, utile per ricoprire distanze al di sopra dei 1000 metri. Questo circuito deve essere alimentato con una tensione variabile dai 3,3V ai 5V. Tali caratteristiche rendono questo circuito utile per applicazioni di audio e video diffusione a distanza.

RS supporta la prototipazione rapida mettendo a disposizione, in esclusiva, la nuova scheda applicativa MBED

RS Components ha annunciato la disponibilità di una nuova scheda applicativa mbed. Basata sulla piattaforma di sviluppo mbed, la nuova scheda applicativa include una serie di connettori e interfacce esterne, non necessita di schede aggiuntive e di conseguenza permette di ridurre considerevolmente i tempi necessari per la progettazione.
La nuova scheda applicativa mbed ha le dimensioni di una carta di credito (54 x 86 mm) ed è stata ideata per essere utilizzata con il modulo microcontrollore mbed NXP LPC1768. Grazie alla sua multifunzionalità, la scheda consente di effettuare un numero elevato di esperimenti e progetti potenziali, minimizzando l'ingombro sulla scheda.
La nuova scheda applicativa offre un'incredibile varietà di caratteristiche come un LCD grafico, un accelerometro a 3 assi, sensore di temperatura, terminali per servomotori, LED a controllo PWM, presa ZigBee, connettività Wi-Fi e Bluetooth, connettori Ethernet e USB, un altoparlante e jack I/O audio.

CODICE MIP 2834644

GMB HR84: Grifo Mini BLOCK - Housing, 8 Opto-In, 4 Relè

Il GMB HR84 è un modulo da Barra DIN in grado di alloggiare una CPU Mini Modulo del tipo CAN o GMM da 28 pin o da 40 pin. In questa famiglia sono comprese varie famiglie di CPU come quelle Core 8051, AVR, PIC, ecc. Il GMB HR84 ècompleta di 8 ingressi, galvanicamente isolati e visualizzati tramite LED, 4 uscite a Relè visualizzati tramite LED, uscita RTC visualizzata tramite LED, connettore AMPMODU II, da 4+4 vie, per comunicazioni seriali, alimentatore stabilizzato, linee di A/D, PWM, D/A, ecc. In un ingombro minimo, contiene un completo controllore in grado di gestire completamente una specifica applicazione.
L' ingombro del GMB HR84 è: frontale 90x71 mm, altezza 58 mm, montaggio su barra DIN ad Omega.
Il GMB HR84 è distribuito da Grifo - Via dell'Artigiano 8/6 - 40016 San Giorgio di Piano (BO)

CODICE MIP 2834505

TELEDYNE LECROY ANNUNCIA LA CONSEGNA DEL NETWORK ANALYZER SPARQ AL LABORATORIO DI ELETTRONICA PER MICROONDE DEL POLITECNICO DI TORINO

Teledyne LeCroy annuncia la consegna di uno SPARQ-3004E Signal Integrity Network Analyzer al Politecnico di Torino in Italia. L'università è riconosciuta in tutto il mondo come istituto di elevatissimo livello per l'ingegneria e l'architettura. L'analizzatore di rete sarà utilizzato nel Laboratorio di Elettronica per Microonde, dove ingegneri e ricercatori hanno apportato significativi contributi alla teoria e alla progettazione degli analizzatori di rete sin dal 1982, anno in cui sono cominciati gli studi su tali strumenti. I tanti risultati ottenuti includono la tecnica "active loop" per la regolazione del carico, la teoria generalizzata della "calibrazione multiporta", e la caratterizzazione del sistema differenziale "load-pull".

CODICE MIP 2834009

I microcontroller PIC® USB 8-bit senza quarzo tagliano i costi di sistema e il consumo di potenza

Un clock con precisione allo 0.25% consente connettività USB, eliminando la necessità di un quarzo esterno

I microcontroller USB (MCU) piu' economici e con il più piccolo fattore forma, hanno un pin-count tra 14 e 100 e sono i primi MCU 8-bit con controllo LCD integrato, RTCC sostenuto da batteria, e USB, tutto in un solo chip.

I più recenti MCU USB PIC® di Microchip offrono un clock interno di riferimento con lo 0.25% di precisione per consentire connettività USB senza quarzo esterno. Sono anche i primi MCU USB a combinare pin-count in un range tra 14 e 100, con una elevata integrazione di periferiche e fino a 128 KB di Flash. La tecnologia eXtreme Low Power (XLP) tiene anche il consumo di potenza sotto i 35 µA/MHz in active mode e 20 nA in sleep mode.

Costo più basso e form-factor più piccolo

Gli MCU PIC16F145X ti offrono connettività USB e rilevamento tattile capacitivo, in aggiunta ad un ampia gamma di periferiche integrate con un ingombro ridotto a soli 4x4 mm.

Rilevamento tattile di elevate prestazioni con USB

Con un Charge Time Measurement Unit (CTMU) integrato e funzionamento tra 1.8 V e 5 V, gli MCU PIC18F2X/4XK50 sono pin-compatibili con i precedenti MCU PIC18, offrendo una facile migrazione a più elevate prestazioni.

USB più controllo LCD e un RTCC con Vbat

La famiglia PIC18F97J94 ti offre connettività USB con controllo LCD, e un real-time clock calendar (RTCC) sostenuto da batteria, tutto su un singolo microcontroller PIC® 8-bit.

PRONTI IN SOLI 3 FACILI PASSI:

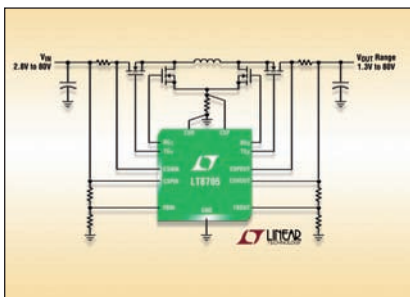
1. Scegli un mix di periferiche e il pin-count che meglio si adattano alla tua applicazione
2. Utilizza gli stack USB gratuiti e i driver software per una più rapida progettazione
3. Inizia a sviluppare con i Kit di sviluppo low-cost

Per maggiori informazioni, vai su: www.microchip.com/get/eu8bitUSB

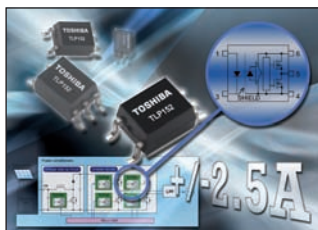
CODICE MIP 2822244

Microcontrollers • Digital Signal Controllers • Analog • Memory • Wireless

Conversione di qualsiasi tensione di ingresso da 2,8V a 80V in tensione o corrente di uscita fisse con efficienza fino al 98% a 250W

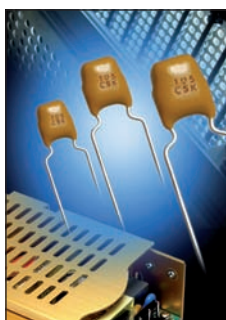
Linear Technology Corporation annuncia l'LT8705, un controller DC/DC buck-boost sincrono ad altissima efficienza (fino al 98%) che funziona con tensioni di ingresso superiori, inferiori o uguali alla tensione di uscita regolata. Questo dispositivo è dotato di quattro loop di feedback per regolare la corrente/tensione di ingresso e quella di uscita. I loop di feedback della corrente e della tensione di ingresso possono impedire il sovraccarico delle celle solari. Il loop della corrente di uscita fornisce una corrente di uscita regolata per il caricabatteria o una fonte di corrente. L'LT8705 funziona in un ampio range di tensioni di ingresso da 2,8V a 80V e produce una tensione di uscita da 1,3V a 80V utilizzando un singolo induttore con un rettificatore sincrono a 4 switch. Un singolo dispositivo può fornire potenza di uscita fino a 250W. La disposizione in parallelo di più circuiti garantisce una maggiore potenza. La frequenza operativa è selezionabile da 100kHz a 400kHz e può essere sincronizzata su un clock esterno. L'LT8705 si basa su un'architettura di controllo in current mode proprietaria per il funzionamento a frequenza costante in modalità buck e boost e utilizza onboard quattro potenti driver per i gate dei MOSFET a canale N.

CODICE MIP 2834707

L'accoppiatore Toshiba per il pilotaggio di IGBT/MOSFET riduce i consumi e lo spazio occupato sulla scheda del 50%

Toshiba Electronics Europe ha presentato un nuovo fotoaccoppiatore che riduce lo spazio occupato sulla scheda e l'assorbimento energetico dei circuiti che utilizzano IGBT e MOSFET dove è richiesto un isolamento galvanico. L'accoppiatore per pilotaggio di gate TLP152 è stato progettato per poter pilotare IGBT e MOSFET di potenza senza alcun componente aggiuntivo.
Un contenitore miniaturizzato in formato S06 riduce lo spazio occupato sulla scheda del 50% rispetto ai comuni componenti in contenitore SDIP. Inoltre, abbassando la tensione di alimentazione minima a 10 V, il nuovo accoppiatore contribuisce anche a ridurre il consumo energetico.
Il contenitore avanzato SO6 di Toshiba garantisce una tensione di isolamento minima di 3750 Vrms ed è compatibile con i requisiti dell'isolamento rinforzato richiesti dalle norme internazionali sulla sicurezza. Il fotoaccoppiatore TLP152 è compatibile con un'ampia gamma di tensioni di ingresso da 10 V a 30 V e minimizza la necessità di inserire circuiti di conversione di potenza supplementari. La corrente massima assorbita è di 3 mA. Un buffer logico con uscita in configurazione totem pole può erogare una corrente di picco massima di +/-2,5 A.

CODICE MIP 2834712

I NUOVO CONDENSATORI SKYCAP CON QUALIFICA AEC-Q200 DI AVX SONO IDEALI PER LE APPLICAZIONI AUTO MOTIVE AD ALTA TENSIONE

AVX Corporation ha sviluppato un'esclusiva e nuova serie di condensatori MLC ad alta tensione, con rivestimento conforme e fili radiali, mirati alle applicazioni dell'industria automotive. Un'estensione della riuscita serie Skycap, i condensatori X7R dielettrici ad alta tensione della nuova serie AR di AVX presentano una gamma delle tensioni di funzionamento compresa tra 1000 V e 3000 V, detengono la qualifica AEC-Q200 e sono subito disponibili nei due diffusi tipi di dimensioni AR21 e AR30. I nuovi condensatori Skycap ad alta tensione serie AR di AVX sono dotati di costruzione robusta, eccellenti caratteristiche di resistenza alle vibrazioni e una notevole resistenza termica, fattori che determinano l'alta qualità per cui la serie Skycap è nota.

I valori di capacitanza della serie AR vanno da 470 pf 68.000 pf, mentre i valori di tolleranza sono: ±5%, ±10%, ±20% e +80%-20%. Interamente qualificata secondo le specifiche AEC-Q200 del settore automotive, la serie AR Skycap è idonea all'impiego con l'elettronica per automotive, il disaccoppiamento e la filtrazione dell'alimentazione e a numerose applicazioni dell'alta tensione.

CODICE MIP 2834714

"RoboBusiness Europe" la prima manifestazione europea dedicata alla robotica business-to-business

La prima edizione europea della manifestazione per lo sviluppo dell'industria europea della Robotica si terrà al Centro Congressi Porto Antico di Genova l'11 e il 12 Aprile. Aprirà l'evento una general overview sul mercato attuale e potenziale della robotica in Europa, seguita poi da sessioni specialistiche: robotica medica, industriale, robot per soccorso, sicurezza e ambienti pericolosi, Venture Capital-Financing. Autorevoli esperti internazionali hanno già confermato la presenza alla Conference, tra cui i rappresentanti di: International Federation of Robotics (IFR), European Factories of the Future Research Association (EFFRA), il Karlsruher Institut für Technologie (KIT), l'Istituto Italiano di Tecnologia (IIT), l'Associazione Italiana di Robotica e Automazione (SIRI). Anche la Commissione Europea, nella Unit Robotics della DG CONNET, che rappresenta il massimo livello della Commissione per la robotica, ha già confermato la partecipazione a quello che vuole essere il "momento storico" verso il primo network europeo della robotica. RoboBusiness Europe adotta il format americano della "Conference & Expo": un susseguirsi di tavole rotonde, meeting, talkshow, networking, condivisione di esperienze e case history internazionali finalizzati al business.

CODICE MIP 2834660

Arrow Electronics annuncia un accordo di distribuzione con Microchip Technology

Arrow Electronics, Inc. ha annunciato oggi che l'ampia gamma di semiconduttori embedded e soluzioni di Microchip Technology Inc. è ora disponibile nell'ambito dell'offerta di componenti di Arrow, sulla base di un accordo globale di distribuzione tra le due società.
Questo accordo comprende l'intera linea di microcontrollori PIC a 8, 16 e 32 bit di Microchip, i semiconduttori analogici e di interfaccia, le soluzioni wireless e i dispositivi di memoria, con i relativi tool di sviluppo. Arrow è anche un distributore primario dei prodotti di SMSC, recentemente acquisita da Microchip.

CODICE MIP 2833682

RS introduce il nuovo Raspberry PI Model A

RS Components ha annunciato che avvierà a breve la distribuzione del nuovo Raspberry Pi Model A, il computer low cost delle dimensioni di una carta di credito sviluppato per incentivare la diffusione della programmazione tra i giovani e gli appassionati di tutto il mondo. RS ha inoltre annunciato di aver siglato un nuovo accordo di distribuzione con la Fondazione Raspberry Pi per la vendita di Raspberry Pi in tutto il mondo. Con un prezzo di poco superiore ai 20 Euro (20,46 per la precisione), il modello A di Raspberry Pi offre una RAM di 256MB, un'unica porta USB e, diversamente dalla versione B, non dispone della connessione Ethernet. Di conseguenza, oltre ad una riduzione dei costi, il nuovo Model A ha un profilo più basso e consuma decisamente meno della versione B, rendendolo quindi la soluzione adatta per applicazioni alimentate a batteria come avviene nell'ambito della robotica, della registrazione dati e nella creazione di stazioni multimediali domestiche a costo zero. Si tratta di una soluzione ideale anche per il settore dell'automazione industriale e del controllo, ambiti che richiedono tipicamente poche caratteristiche integrate per il loro sviluppo.

CODICE MIP 2834736

IDT annuncia i primi oscillatori MEMS ad alte prestazioni

Integrated Device Technology ha annunciato i primi oscillatori MEMS ad alte prestazioni a quattro frequenze e uscite sincrone multiple. I più recenti oscillatori di IDT offrono uscite configurabili in base alla proprie esigenze in un contenitore compatto compatibile con le soluzioni standard più diffuse sul mercato, che consente di risparmiare spazio sulla scheda e ridurre i costi della componentistica per applicazioni nel settore delle telecomunicazioni, reti di trasmissione dati, archiviazione, controllo industriale e FPGA.
Gli oscillatori MEMS evoluti della serie 4E da ±50 ppm di IDT integrano un uscita LVDS o LVPECL con un'uscita sincrona CMOS all'interno dello stesso contenitore, eliminando la necessità di ricorrere ad un cristallo esterno o a un oscillatore secondario. Disponibili per frequenze fino a 600 MHz, questi nuovi oscillatori fanno risparmiare spazio sulla scheda, semplificano il circuito applicativo e riducono il costo nelle applicazioni ad alte prestazioni che richiedono sorgenti di frequenza di tipo LVDS o LVPECL. Inoltre, gli oscillatori dispongono di due piedini di controllo per scegliere tra le quattro frequenze di uscita programmabili in fabbrica, permettendo di rimpiazzare quattro componenti diversi con un singolo dispositivo. Questa caratteristica permette ai clienti di ridurre i componenti inseriti nella distinta base, di ridurre le scorte di magazzino e di risparmiare sui costi.

CODICE MIP 2834745

LEM aggiunge trasduttori di precisione da 130A e 150A

LEM ha appena introdotto sul mercato la serie LA 130-150 di trasduttori di corrente. Questi dispositivi montati su PCB condividono un footprint comune molto compatto e la disposizione dello schema del case e sono disponibili per misurazioni con fondo scala nominale di 130 e 150 ARMS. Le serie LA 130-150 misura correnti DC, AC, e pulsate, offrendo un output sotto forma di un segnale di corrente proporzionale alla corrente (primaria) misurata. Due i range di conversione, un fattore 1000 o 2000, offerti, con due stili di montaggio; i trasduttori possono essere forniti con una piena apertura attraverso cui il conduttore che trasporta la corrente primaria è fatto passare, oppure possono essere forniti pronti all'uso, con un conduttore primario che si connetterà direttamente al PCB ospite. La serie di trasduttori LA 130-150 offre una banda passante massima tra DC e 150 kHz (-1dB point) ed una precisione migliore dello 0.5% sulla corrente primaria nominale RMS e a temperatura ambiente. Errore di linearità tipico di ±0.1% alla corrente nominale e la massima deriva termica è uno ±0.46% di corrente nominale tra -40 e +85°C. La tecnologia effetto Hall Closed-Loop consegue una perdita di inserzione pari a zero e d un tempo di risposta molto veloce, inferiore a 500 nsec.

CODICE MIP 2834749

o GENIUS?

elettronica con Inware Edizioni!

MAKER MEMBERSHIP

Special Price € 49,50

- 11 nuovi numeri di Fare Elettronica in PDF (in coda all'eventuale abbonamento esistente)

Bonus Pack MAKER che comprende l'accesso e/o download:

- agli articoli (centinaia) del CLUB di Fare Elettronica
- un ebook a scelta* (Smart card, CPLD, display LCD, PIC, AmpOP, alimentatori, linguaggio C, Basic per PIC, Elettronica Analogica)
- una raccolta delle annate di Fare Elettronica in PDF a scelta* (dal 2003 al 2012)

€ 39,50

SMART MEMBERSHIP

- tutto l'archivio delle riviste firmware pubblicate mensilmente a partire dal febbraio 2010
- 11 nuovi numeri della rivista digitale firmware (in coda all'eventuale abbonamento esistente)

Bonus Pack SMART che comprende l'accesso e/o download:

- a tutto l'archivio delle riviste firmware pubblicate mensilmente a partire dal febbraio 2010
- agli articoli (centinaia) del CLUB di Firmware
- un ebook a scelta* (Smart card, CPLD, display, PIC, AmpOP, alimentatori, linguaggio C, Basic per PIC, Elettronica Analogica)
- una raccolta della annate di Firmware in PDF (dal 2006 al 2012) a scelta*

SI, VOGLIO RICEVERE SUBITO LA MIA MEMBERSHIP CARD

☐ MAKER MEMBERSHIP ☐ SMART MEMBERSHIP ☐ GENIUS MEMBERSHIP

METODI DI PAGAMENTO

☐ **BONIFICO BANCARIO** ☐ **BOLLETTINO POSTALE** ☐ **ALLEGO ASSEGNO** (Intestato a Inware Edizioni)

☐ MasterCard ☐ VISA ☐ postepay

TITOLARE. . . . .

NUMERO . . . .

DATA DI SCADENZA. . . . . CODICE DI SICUREZZA . . . .

PayPal fare il pagamento su paypal@inwaredizioni.it

INVIARE IL MODULO D'ORDINE A: Inware Edizioni srl, via Giotto 7 - 20032 Cormano (MI) - oppure via FAX allo 02 66508225

(Inviando questo modulo si accettano i termini e le condizioni riportate a tergo)

Nome . . . . Cognome . . . .

Indirizzo . . . . n° . . . .

Cap . . . . . . Città . . . . Prov . . . .

Tel . . . . Fax . . . . Email . . . .

Ragione Sociale . . . . Codice fiscale . . . .

Partita IVA . . . .

DATA . . . . FIRMA . . . . ☐ Fattura

Fiera del Radioamatore di Pordenone

Fiera del Radioamatore: radiantistica, componenti e ricambi per apparecchiature radio, pubblicazioni tecniche, antenne per ricetrasmettitori, elettronica di consumo, prodotti per telecomunicazioni, informatica.

Dove: Pordenone
Quando: 27-28 Aprile 2013
Orari: dalle 9.00 alle 18.00
Organizzazione: Fiera di Pordenone
info:www.radioamatorepordenone.it

CODICE MIP 2805239

Ham Radio

Un appuntamento storico per i radioamatori. L'Ham Radio di Friedrichshafen è la fiera europea di riferimento per radioamatori ed hobbisti elettronici che attrae visitatori da tutto il mondo. In occasione della fiera si terranno workshop e conferenze di sicuro interesse per gli appassionati del settore.

Dove: Friedrichshafen (Germania)
Quando: 28-30 Giugno 2013
Orari: dalle 9.00 alle 18.00
Organizzazione: Messe Friedrichshafen
info: www.hamradio-friedrichshafen.de

CODICE MIP 2806570

EXPOELETTRONICA - FORLÌ

Expo Elettronica ha un pubblico vasto ed eterogeneo: appassionati del "fai da te", elettro-riparatori, "smanettoni", radioamatori, "cacciatori" di buone occasioni o pezzi rari; questo perché propone un panorama merceologico e un calendario di eventi collaterali veramente ricchissimo. Al suo interno si svolge uno degli appuntamenti più caratteristici ed atteso: il mercatino dell'usato.

Dove: Forlì
Quando: 4-5 Maggio 2013
Orari: dalle 9.00 alle 18.00
Organizzazione: Blunautilus
info:www.expoelettronica.it

CODICE MIP 2799372

MARC di Primavera

Fiera dell'elettronica, radiantistica, surplus, telefonia. Grandi occasioni anche per informatica, CD, DVD e stampa specializzata.

Dove: Genova
Quando: 11-12 Maggio 2013
Orari: dalle 9.00 alle 18.00
Organizzazione: Studio Fulcro
info:www.studiofulcro.it

CODICE MIP 2799411

Mostra-mercato di apparati e componenti per telecomunicazioni, Internet e ricetrasmissioni di terra e satellitari, antenne, elettronica, informatica, telefonia statica e cellulare, hobbistica ed editoria; borsa-scambio di surplus radioamatoriale, informatico ed elettronico). Attualmente RADIANT, che si sviluppa su una superficie espositiva superiore ai 10.000 mq. coperti, con più di 180 espositori per edizione ed oltre 40.000 visitatori annui, è considerata la più importante Mostra-Mercato nazionale del suo settore. Fra le categorie merceologiche ricordiamo: elettronica, informatica, telefonia, radiantismo, editoria, tv satellitare, hobbistica, surplus, radio d'epoca.

Dove: Novegro (MI)
Quando: 1-2 Giugno 2013
Orari: dalle 9.00 alle 18.00
Organizzazione: Comis
info:www.parcoesposizioninovegro.it

CODICE MIP 2806564

GRANDE FIERA DELL'ELETTRONICA

ERBA (CO)

presso Lariofiere

CODICE MIP 2831685

13-14 APRILE 2013

orario continuato dalle 09.00 alle18.00

Fiera dell'Elettronica in Genere

BUONO SCONTO DI € 2,00

Da presentare alla biglietteria (non cumulabile)

Fiera Elettronica - ERBA (CO) 13-14 APRILE 2013

www.italfiere.net

Puoi scaricare lo sconto sul biglietto d'ingresso:

Per ricevere lo sconto alle prossime edizioni compila questo modulo:

Email: ....

Provincia di provenienza: ....

Come hai saputo di questa fiera? ....

di GIUSEPPE LA ROSA

WEB SERVER A 4 CANALI

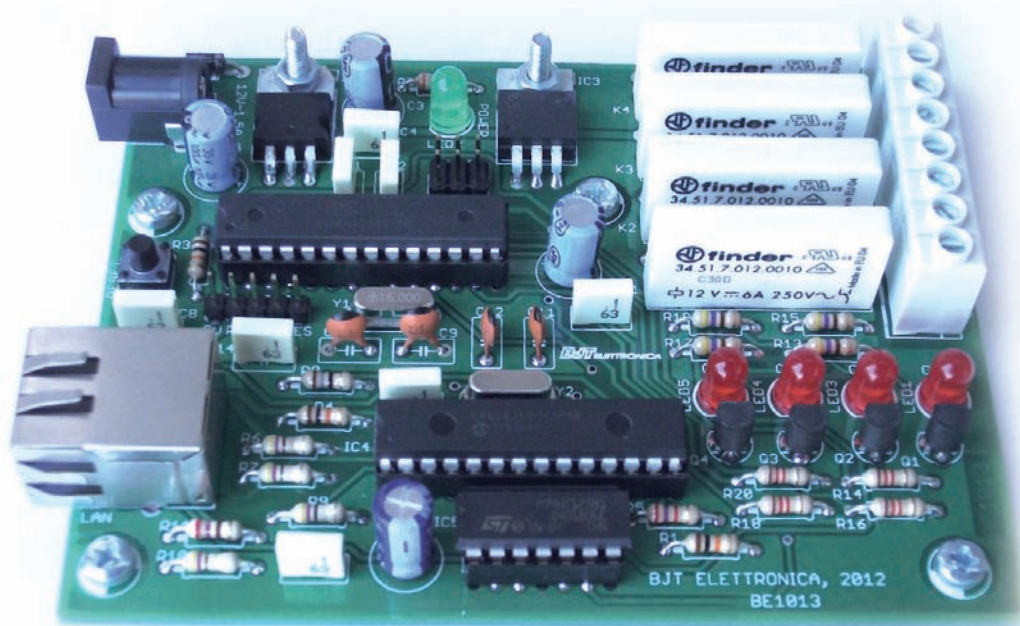



Figura 1: Foto del progetto finito.

In **figura 1** è visibile il prototipo della scheda di comando su rete **LAN** o **Internet** completamente basata su **ARDUINO UNO** con la possibilità attraverso un **PC** o **Tablet** di interagire su apparati elettrici, mediante quattro relè da **6 Ampere 250 Volt** con contatto normalmente aperto.

Le applicazioni sono molteplici e varie: spegnimento/accensione di caldaie oppure luci e fari interni ed esterni, ancora apertura di cancelli automatici ad esempio da computer remoto (segreteria o portineria).

Abbinato ad una bobina di sgancio è possibile togliere la corrente da qualsiasi elemento elettrico, utilizzando un **RESTART** ripristina la corrente.

Abbinato ad un videoregistratore **Web Server** è possibile verificare in **Real-Time** l'esecuzione dei comandi da remoto.

Stando comodamente seduti al **PC**, la scheda consente inoltre, di interagire sul qualsiasi dispositivo evitando di spostarsi per attivare e disattivare un apparecchio elettrico.

SCHEMA ELETTRICO

Partiamo con l'analisi dello schema elettrico visibile in **figura 2**.

Il cuore principale del **Web Server** e il microcontrollore **IC2** (**ATMEGA328P**) che dirige tutte le funzioni operative, come la comunicazione su **Bus SPI** (piedini **4**, **16**, **17**,**18** e **19**) e il controllo dei **Relè** (**K1**, **K2**, **K3**, e **K4**) tramite i **Transistor** in configurazione di interruttore elettronico **Q1**, **Q2**, **Q3** e **Q4** rispettivamente connesse ai piedini **11**, **12**, **13**, e **14**. Il connettore **J1** consente la programmazione **ICSP** del microcontrollore **IC2**, mentre **J3** implementa un'interfaccia seriale **UART** per la programmazione del microcontrollore previo caricamento del **Bootloader**.

Il quarzo **Y1** fornisce la frequenza di **Clock** da **16 MHz** necessaria al funzionamento **IC2**. Il microcontrollore **IC4** della **Microchip** svolge le funzioni di interfacciamento con **IC2** e di conversione dei dati secondo il protocollo **Ethernet**; il microcontrollore è l'**ENC28J60** e contiene una completa interfaccia **Ethernet** di tipo **10 BaseT**, conforme allo standard **IEEE 802.3** ed è interfacciata tramite **Bus SPI** (**Pin** di

Realizziamo un economico Web Server basato su ARDUINO UNO con Data-Rate fino 10Mbps per applicazioni domotiche.

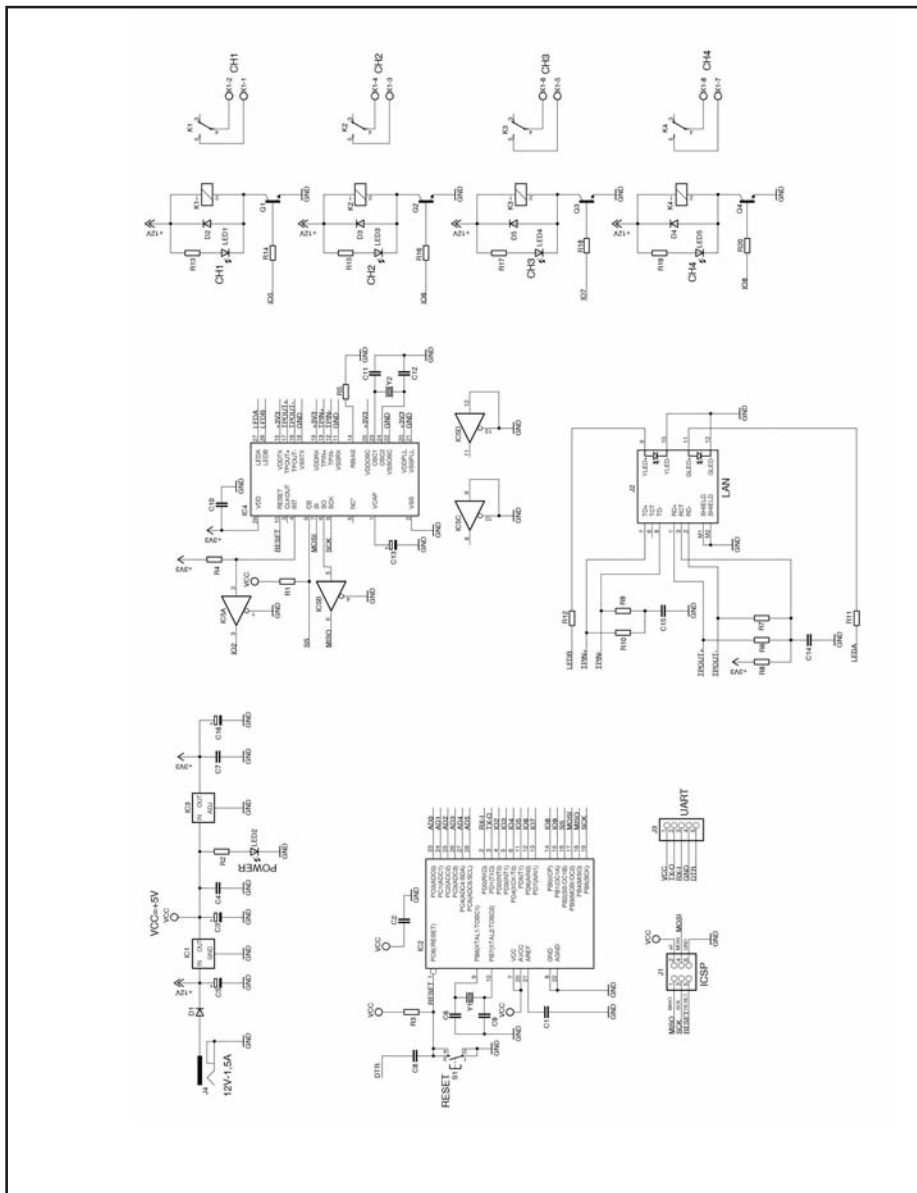



Figura 2: Schema elettrico della scheda Web Server.

IC4 4, **6**, **7**, **8** e **9**) configurata per un **Clock** massimo di **20 MHz**.
IC5A e **IC5B**, servono ad adattare i livelli logici con cui lavora **IC4** (che sono **0** e **3,3 Volt**) a quelli di **IC2** (**0** e **5 Volt**).
Notate che le altre linee di **IC4** interfacciate con **IC2** (piedini **7**, **8**, **9** e **10**) non richiedono il **Buffer IC5**, perché sono speciali **Pin** di **Input** dell'**ENC28J60** compatibile con logica a **5 Volt**; in pratica possono accettare segnali **0** e **5 Volt** anche se **IC4** è alimentato a **3,3 Volt**. Il pulsante di **RESET S1** collegato in entrambi microcontrollori **IC2** e **IC4** permette di resettare l'interfaccia **Ethernet IC4** e il microcontrollore **IC2**, all'occorrenza in modo manuale. La presa **RJ45 J2** ha due coppie di linee per la trasmissione (**TPOUT+** e **TPOUT**) e ricezione (**TPIN+** e **TPIN**-) dei dati di **IC4** ed altre due linee riservate al controllo dei **LED** di stato (**LEDA** e **LEDB**) integrati in essa, che servono ad indicare la presenza del **Link Ethernet** e l'altro il trasferimento dei dati. All'interno della presa **J2** la coppia di linee per la trasmissione (**TPOUT+** e **TPOUT**-) termina sul primario di un trasformatore d'accoppiamento a presa centrale polarizzata da **3,3 Volt** mediante la resistenza **R8**, che termina sul piedino **3** di **J2,** comune dei primari.
In parallelo ad ogni sezione del primario si trovano una resistenza **R6** e **R7**.
Le linee d'ingresso (**TPIN+** e **TPIN-**) sono connesse, invece, al secondario di un trasformatore a secondario semplice ed entrambe sono filtrate verso massa dalle resistenze **R9** ed **R10** e dal condensatore **C15**. Il condensatore **C13** connesso al piedino **1** di **IC4** filtra la tensione d'uscita a **2,5 Volt** del regolatore integrato all'interno. La resistenza **R5** serve per la polarizzazione del **Transceiver LAN** che fa capo alle linee di recezione e di trasmissione dati. L'**ENC28J60 IC4** funziona con **Clock** di **25 MHz** che è quello consigliato per garantire la connettività a **10 Mbps**, definito dal quarzo **Y2** collegato fra i piedini **23** e **24**.
In conclusione dell'analisi dello schema elettrico della scheda passiamo l'alimentazione, essa giunge dal connettore **J4** e tramite il diodo di protezione **D4** va ad alimentare i relè **K1** a **K4** inseguito viene stabilizzata a **5 Volt** tramite il regolatore **IC1** esso fornisce **5 volt** stabilizzati al microcontrollore **IC2** e al Buffer **IC5**, essa è resa disponibile anche sui connettori **J1** e **J3**.
Lo stabilizzatore **IC3** fornisce **3,3 Volt** stabilizzati all'interfaccia di rete **IC4.**

REALIZZAZIONE PRATICA

Passiamo adesso alla costruzione della scheda che si presenta abbastanza semplice, la basetta è del tipo a doppia faccia con fori metallizzati e si prepara a partire dalle due tracie di **figura 4** e **figura 5**, rispettivamente lato saldature e lato componenti.
Ottenuto il circuito stampato montate i componenti richiesti dalla **lista compo-**

Figura 3: Piano di montaggio.

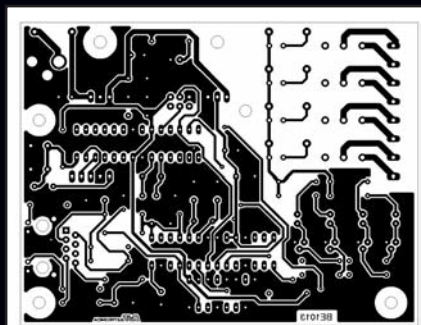



Figura 4: Circuito stampato in scala 1:1 (lato saldature).

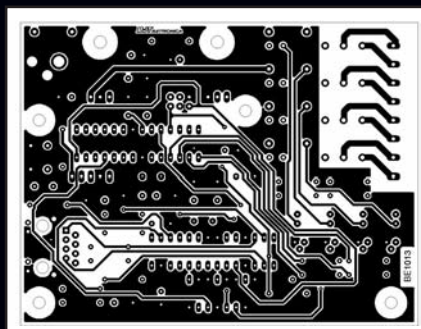



Figura 5: Circuito stampato in scala 1:1 (lato componenti).

nenti, partendo dalle resistenze dai diodi e gli zoccoli per gli integrati, quindi proseguendo con i condensatori prima i non polarizzati e poi gli elettrolitici; poi è la volta dei quarzi **Y1** e **Y2** che devono essere saldati a **2 mm** dalla piastra per evitare cortocircuiti.
Continuate con il montaggio dei transistor **Q1** a **Q4** e dei diodi **LED**.
Montate i due regolatori **IC1** e **IC3** che vanno saldati sdraiati e isolati tramite mica per **TO220** e avvitati al circuito stampato tramite viti **3 mm** più dadi.
Completate il montaggio inserendo e saldato il pulsante miniatura per il **RESET S1**, la presa **RJ45 J2** quindi i due strip maschi **J1** e **J3**, i morsetti **X1** e il connettore di alimentazione **J4**.
A questo punto non vi resta che inserire gli integrati, orientarli come mostrato nel piano di montaggio di **figura 3**.
Terminato e verificato il montaggio, per passare alla fase successive bisogna caricare **Bootloader** con apposito programmatore che va collegato al connettore **J1**.
Se si dispone di un **ATMEGA328P** già con **Bootloader** caricato si può passare al caricamento del **Firmware**, operazione descritta nel prossimo paragrafo.

PROGRAMMAZIONE DELLA SCHEDA

Per modificare tutti i parametri della scheda come: **IP**; **MAC**; **PORTA TCP**; **DDNS**; **NOME**; **PASSWORD**.
Scaricate il **Sorgente** e la **Libreria** per **ARDUINO UNO** dal sito di **Fare Elettronica**.
Installate la **Libreria** nell'ambiente **IDE ARDUINO UNO**.
Nel sorgente (vedi **Listato 1**) al punto **1** va inserito un indirizzo **MAC** che può essere preso da una vecchia scheda di rete non usata, é anche possibile lasciare quello già presente nel **Sorgente**.
Nel punto **2** va inserito l'**IP** che intendete usare separato da virgole.
Al punto **3** va inserita la porta pubblica è consigliato lasciarla invariata (valore **80**).

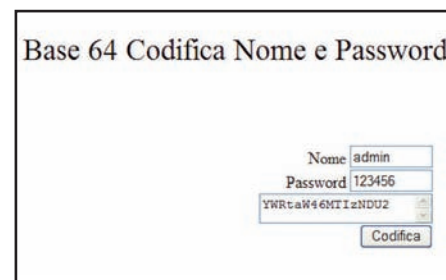



Figura 6: Pagina Web per la codifica della Password e Nome in Base 64.

LISTATO 1 Parte del Sorgente da modificare.

```
// Nella vostra rete locale. Non si può avere lo stesso MAC e IP in due dispositivi

uint8_t mymac[6] = {0xBE,0xBA,0x50,0x2C,0x6F,0x67}; //-1 Indirizzo MAC MODIFICA
uint8_t myip[4] = {192,168,1,200};                  //-2 Indirizzo IP  MODIFICA
uint8_t mymask[4] = {255,255,255,0};
uint16_t MYWWWPORT = 80;                  //-3 Porta tcp (max gamma 1-254) consigliata 80
MODIFICA
char baseurl[]="http://esp.homepc.it/";  //-4 Inserire DDNS (o IP di rete) max 29 caratteri
MODIFICA
                                        //   esempio http://esempioprov.homepc.it/
                                        //   Servizio DDNS http://dyndns.it/
char pass[]="YWRtaW46MTIzNDU2";           //-5 Nome e Password generare la codifica base 64
MODIFICA
                                        // Nome demo: admin
                                        // Password demo: 123456
#define BUFFER_SIZE 750
static uint8_t buf[BUFFER_SIZE+1];
```

Nel punto **4** va inserito **Host Name** di un massimo di **29** caratteri se si utilizza un servizio di **DDNS** altrimenti se si impiega in una rete locale inserire l'**IP** che avete assegnato in precedenza.
La **Password** e il **Nome** prima devono essere codificate in **Base 64**, quindi scegliere un **Nome** di **5** caratteri e una **Password** di **6** caratteri.
Sempre dal sito di **Fare Elettronica** scaricate il software per la conversione in **Base 64** in formato di pagina **Web** (vedi **figura 6**).
Inserire nella casella il **Nome** scelto e nell'altra casella la **Password** che volete adoperare, cliccate su codifica e sulla casella sottostante verrà visualizzata una sequenza di caratteri che corrispondono alla vostra **Password** e **Nome** codificate in **Base 64**.

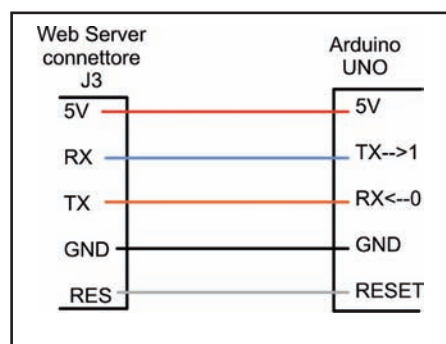



Figura 7: Collegamento tra ARDUINO UNO e il Web Server per la programmazione.

Copiate la sequenza di caratteri ed incollatela nel sorgente al punto **5** tra virgolette.
Attuate le modifiche descritte in precedenza potete salvare e caricare il **Firmware** sulla scheda.
Servendovi della scheda **ARDUINO UNO** togliendo il microcontrollore, ed operando il collegamento di **figura 7** potete scaricare il **Firmware** al **Web Server**.

COLLEGAMENTO DELLA SCHEDA

Per utilizzare il **Web Server** (vedi **figura 8**) occorre alimentare la scheda con un ali-

LISTA COMPONENTI

Sigla	Valore
R1	10 KΩ 1/4 W
R2	330 Ω 1/4 W
R3÷R4	10 KΩ 1/4 W
R5	2,7 KΩ 1/4 W
R6÷R7	47 Ω 1/4 W
R8	18 Ω 1/4 W
R9÷R10	47 Ω 1/4 W
R11	270 Ω 1/4 W
R12÷R13	270 Ω 1/4 W
R14	1,2 KΩ 1/4 W
R15	470 Ω 1/4 W
R16	1,2 KΩ 1/4 W
R17	470 Ω 1/4 W
R18	1,2 KΩ 1/4 W
R19	470 Ω 1/4 W
R20	1,2 KΩ 1/4 W
C1÷C4	100 nF poliestere
C5	100 µF 25 V elettrolitico
C6	22 pF ceramico
C7÷C8	100 nF poliestere
C9	22 pF ceramico
C10	100 nF poliestere
C11÷C12	15 pF ceramico
C13	10 µF 25 V elettrolitico
C14÷C15	100 nF poliestere
C16	100 µF 25 V elettrolitico
D1÷D5	1N4007 diodo
LED1	Led 5 mm rosso
LED2	Led 5 mm verde
LED3÷LED5	Led 5 mm rosso
Q1÷Q4	BC547B
IC1	L7805CV
IC2	ATMEGA328P
IC3	LD1117T3.3
IC4	ENC28J60-I/SP
IC5	74HC125N
J1	Strip doppio M.6 poli
J2	Connettore RJ45 RJ45EM
J3	Strip M. 5 poli
J4	Presa DC 5,5X2,1 mm
K1÷K4	Relè 12 V-6 A
S1	Pulsante
X1	Morseti a 8 poli
Y1	16 MHz quarzo
Y2	25 MHz quarzo
N.2	Zoccolo 14+14 pin
N.1	Zoccolo 7+7 pin
N.2	Mica per TO220
N.2	Viti 3x16 mm
N.2	Dadi 3 mm

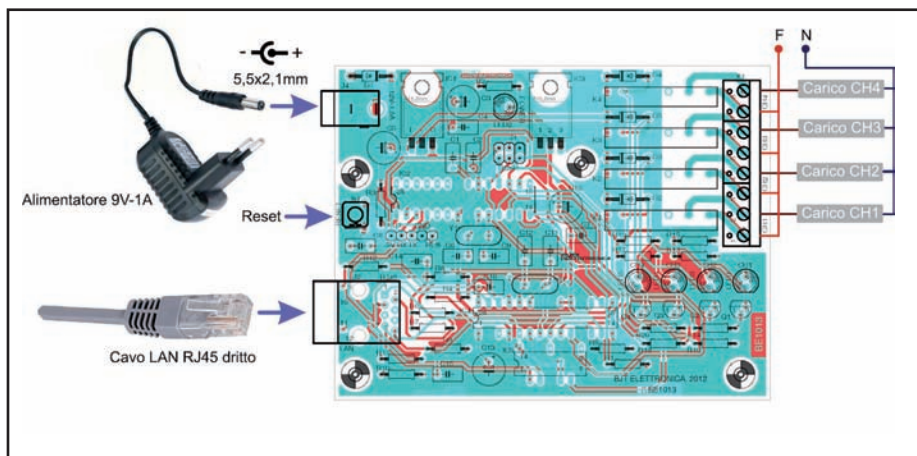
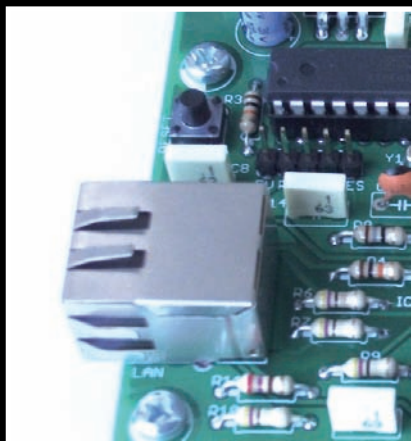



Figura 8: Schema di collegamento.

mentatore da **9 Volt** con corrente di **1 Ampere** provvisto di uno spinotto delle dimensioni di **5,5x2,1 mm** con il positivo centrale e negativo esterno.
Con un cavo di rete dritto bisogna connettere la scheda allo **Switch** o **Modem Router**.
I carichi vanno collegati alla morsettiera **X1** con cavi adeguati all'assorbimento dei carichi, la massima corrente applicabile per singolo morsetto è di **6 Ampere**.

COLLEGAMENTO ALLA RETE LAN

La scheda programmata con **Firmware** scaricato dal sito (sorgente non modificato) già dispone di un **IP** statico impostato (**192.168.1.200**) (vedi **figura 9**) basta collegarla alla rete per effettuare un test, aprire il vostro **Browser** digitare l'indirizzo **IP** (**192.168.1.200**) e vi verrà chiesto **Nome** (**admin**) e **Password** (**123456**) di seguito avrete accesso alla pagina **Web** del **Web Server**.
Questa scheda può essere connessa (**figura 10**) alla vostra rete anche tramite un **Wi-Fi Bridge** in modo da collegarla wireless senza posa di cavi di rete.
Fin qui abbiamo visto l'accesso da remoto tramite rete locale.
Se volessimo accedere al **Web Server** da **Internet** cioè al di fuori dalla nostra abitazione, dovremmo impostare il **Modem Router** accedendo alla voce **Virtual Server** in cui va inserito l'**IP** della scheda, il valore della **Porta Pubblica** e il protocollo **TCP**.

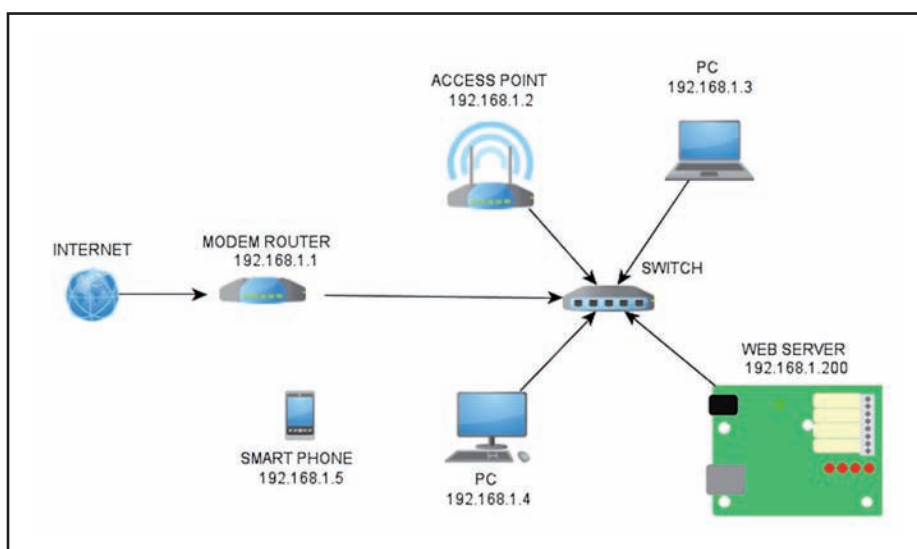



Figura 9: Schema di collegamento alla rete LAN ed a Internet.

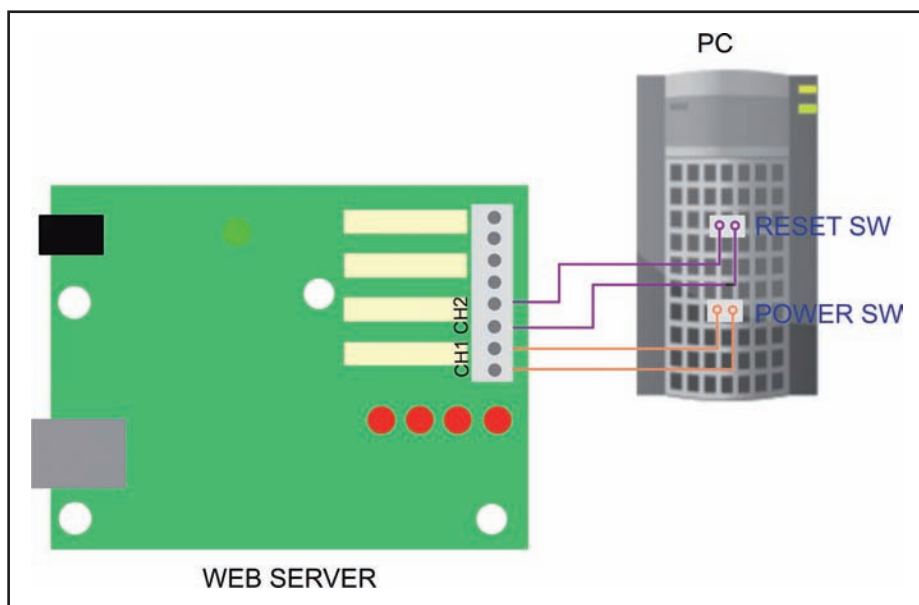



Figura 10: Schema di collegamento tramite Wi-Fi Bridge.

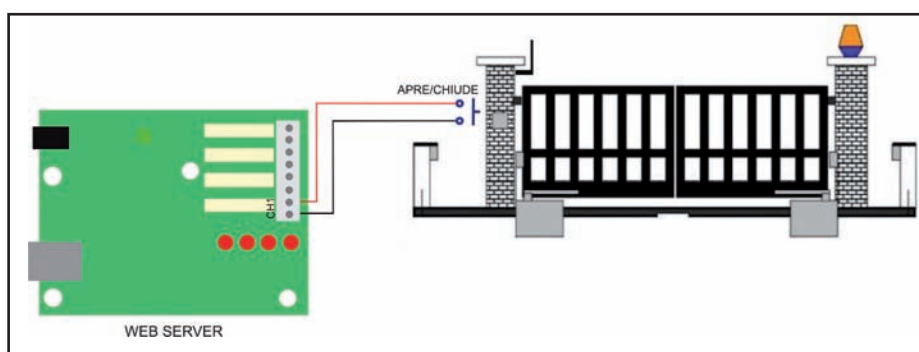



Figura 13: Schema apertura di un cancello automatico.

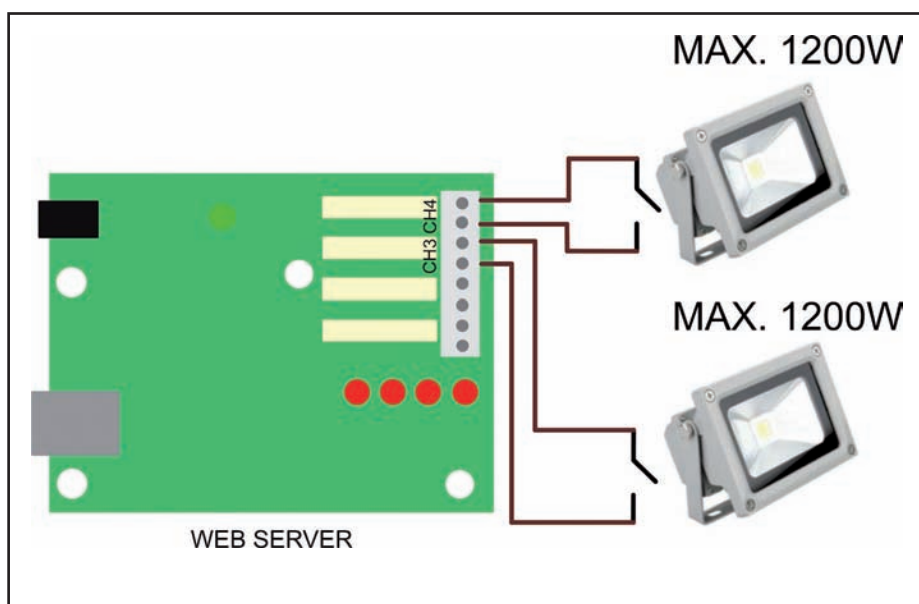



Figura 14: Simulazione presenza tramite accensione di fari.

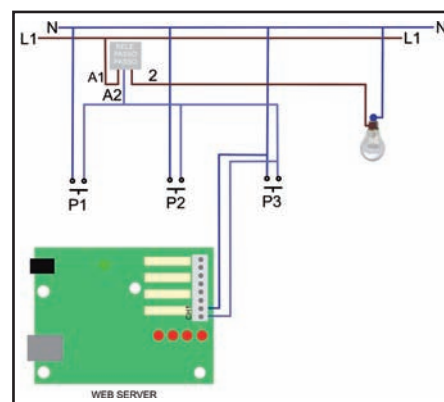



Figura 12: Accensione di una lampada da remoto e da locale.

Se il vostro **IP** pubblico è statico per accedere nel **Web Server** basta digitarlo nel **Browser**, mentre se è dinamico occorre utilizzare un servizio **DDNS** in cui registrare un **Host Name** ad esempio "**esempioprov.homepc.it**", un sito consigliato per la registrazione di un **DDNS** è **http://dyndns.it/**.
Per eseguire l'aggiornamento del vostro **IP** dinamico ogni volta che cambia necessita di impostare dall'apposito menù del vostro **Modem Router** l'aggiornamento automatico al servizio **DDNS** così da aggiornare l'**IP** salvato nell'account del servizio.

APPLICAZIONI DELLA SCHEDA

Lo schema di **figura 11** consente di comandare a distanza tramite Internet il Reset, l'accensione e lo spegnimento di un **PC** collegato alla rete **LAN**.
Il collegamento al computer avviene tramite i canali **CH1** e **CH2** rispettivamente collegati ai terminali **POWER SW** e **RESET SW** della motherboard.
Questa realizzazione permette di accendere il **PC** a distanza e tramite appositi software prelevare file salvati nel computer.
Connettendo la scheda ad un impianto di illuminazione con **relè passo passo**, come mostra la **figura 12**, possiamo accendere e spegnere luci da remoto e tramite i pulsanti da locale.
Questa applicazione non richiede molte modifiche, ma solo l'aggiunta di due cavi e della scheda all'impianto già esistente.
La **figura 13** mostra la connessione ad un

Figura 16: Pagina Web del Web Server visualizzata tramite PC.

cancello automatico già esistente. Basta connettere due cavi al selettore a chiave (contatti Apre/Chiude) e alla scheda per conseguire l'apertura del cancello tramite **Smart Phone**, questo permette di eliminare il radiocomando e di poter aggiungere utenti in caso di cancelli aziendali o condominiali, senza sborsare

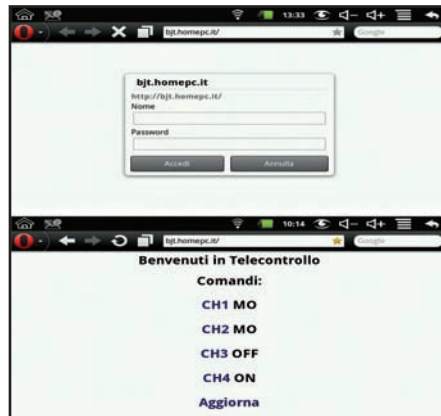



Figura 15: Pagina Web del Web Server visualizzata tramite Tablet.

denari per il radiocomando, fornendo solo **Nome** e **Password**.

La schematizzazione di **figura 14** permette di accendere in modo bistabile due fari che possono essere installati in giardino, inoltre questa configurazione può essere usata come simulazione presenza in casa.

Accendendo e spegnendo i fari a distanza in diversi orari si possono distogliere i ladri al procedere in un eventuale svaligiamento della casa.

Oltre alle applicazioni descritte in precedenza si possono realizzare svariate applicazioni in ambito domestico e industriale ad esempio attivare una pompa fino al riempimento di una cisterna, lasciamo all'utilizzatore la personalizzazione della scheda e l'utilizzo.

BROWSER PC E TABLET

Nelle **figure 15** e **16** possiamo vedere come viene visualizzata la pagina dai **Browser** dei **PC** e dai **Tablet** e **Smart Phone**.

Prima di accedere alla pagina **Web** vengono chiesti **Nome** e **Password** basta inserirli correttamente per accedere alla pagina, cliccando sui primi due **Canali** i carichi vengono accesi in maniera monostabile mentre sui **Canali 3** e **4** i carichi vengono accesi in maniera bistabile (ma questa configurazione può essere modificata da **Sorgente**).

Per l'utilizzo di questo **Web Server** consigliamo di usare per i **PC** il **Browser Google Chrome**, mentre per **Tablet** e **Smart Phone** il **Browser Opera Mobile**.

CONCLUSIONE

Tutti i file per la realizzazione come già è stato detto sono disponibili sul sito di **Fare Elettronica**.

Come avete letto in precedenza le applicazioni non si limitano solo a quelle descritte, ma sono presso che infinite.

Il sorgete è scritto in linguaggio **C** per **ARDUINO UNO** è facilmente modificabile e adattabile alle proprie necessità.

Al seguente link è possibile vedere il **Web Server** in funzione: http://www.youtube.com/watch?v=7iXonypRo-E&feature=player_embedded ❏

CODICE MIP 2835688

di SILVANO BREGGION

PLC con interfaccia USB

(parte seconda)

Riprendiamo l'analisi del nostro PLC descrivendo la sintassi per l'acquisizione delle stringhe di programmazione.

Ogni stringa occupa quattro byte di memoria di tipo EEPROM. Il PIC18F4550 con i suoi 256 byte disponibili, è in grado di memorizzare fino a 63 stringhe. Tutta l'area di memoria EEPROM è fruibile per la loro memorizzazione tranne l'indirizzo 0x00 riservato al puntatore di stringa e gli ultimi 3 Byte, riservati a funzioni speciali attuali e future. Il nostro PLC esegue tutte le istruzioni disponibili in memoria, gestisce gli ingressi e le uscite, visualizza lo stato o il valore di alcuni parametri direttamente sul LCD, definendo questo insieme di azioni col nome di ciclo. Contrariamente a quello che possiamo pensare, la maggior parte del tempo viene usato dal display per visualizzare le informazioni. In ogni caso il tempo necessario per effettuare un ciclo completo è di circa 25 ms, solo quando viene richiesta la visualizzazione dello stato dei Marker, il tempo scende a circa 13 ms.

Per accedere alla fase di programmazione è necessario premere il pulsante P3 (SET) e tenerlo premuto per circa tre secondi. Il display LCD visualizza una stringa vuota il cui numero e quello indicato dal TOP (0x00) incrementato di una unità, tranne nel caso la memoria sia vuota. Il cursore lampeggiante indica in quale campo possiamo incrementare o decrementare il valore o il simbolo. Il pulsante P1 (UP) incrementa il valore mentre il pulsante P2 (DOWN) lo decrementa. Per cambiare campo, sempre da sinistra verso destra, è abilitato il pulsante P3 (SET), mentre per ritornare alla fase operativa in qualsiasi momento è sufficiente premere il pulsante P4 (END).

Terminata la programmazione della stringa, la seconda riga del display mostra la seguente richiesta

SET=save END=trn

il cui significato è piuttosto intuitivo: premendo il pulsante P3 (SET) la stringa viene confermata e salvata in memoria, mentre premendo pulsante P4 (END) viene riproposta la stessa stringa per le necessarie modifiche. Fate bene attenzione a confermare tutte le stringhe che ritenete corrette e controllate ogni volta che numero di stringa sia incrementato di una unità.

Può sorgere la necessità di cancellare una o più righe o addirittura all'interno programma. Sempre all'interno della fase di programmazione selezionate la stringa da cancellare, premette e tenete premuto per circa tre secondi il pulsante P3. La seconda riga del display con la scritta

SET=canc END=rtn

chiede di confermare la richiesta di cancellazione della stringa premendo il pulsante P3 o il ritorno senza azioni con il pul-

SIGLA	COMMENTO	PLC
S1	SENSORE INDUTTIVO LETTORE GIRI	I2
S2	PULSANTE START/STOP MACCHINA	I0
S3	PULSANTE STOP IMMEDIATO	I1
FC1	FINE CORSA 1: TRANCIA IN RIPOSO	I3
FC2	FINE CORSA 2: TAGLIO CONCLUSO	I4
KA1	RELE' START/STOP MACCHINA	Q0
KA2	RELE' COMANDO TRANCIA	Q1

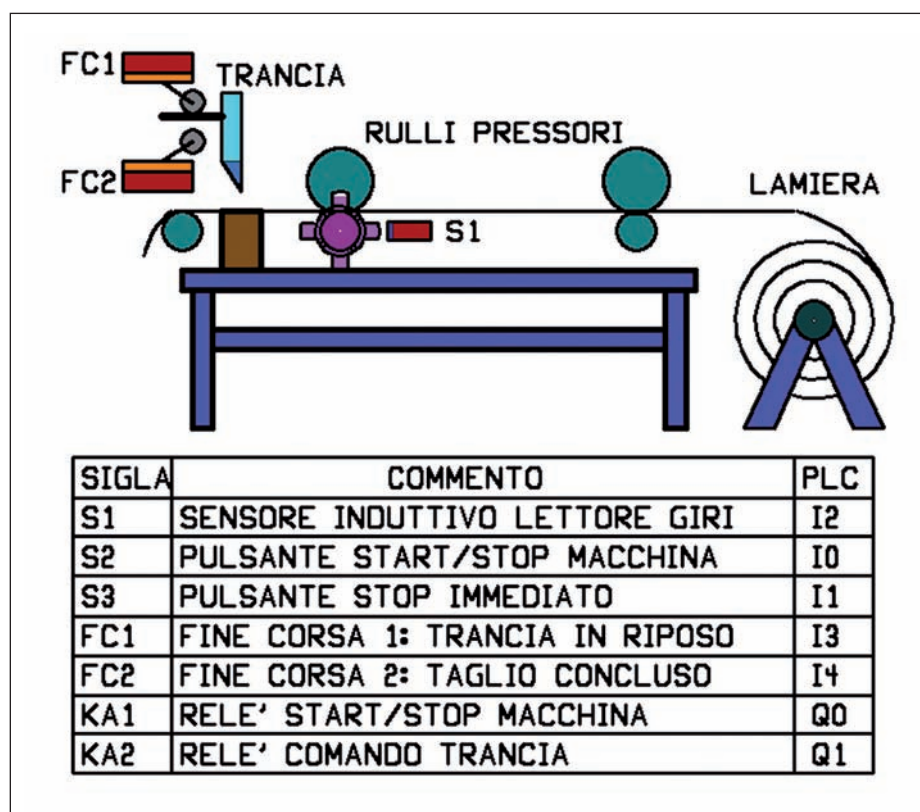



Figura 1: esempio di sbobinatrice

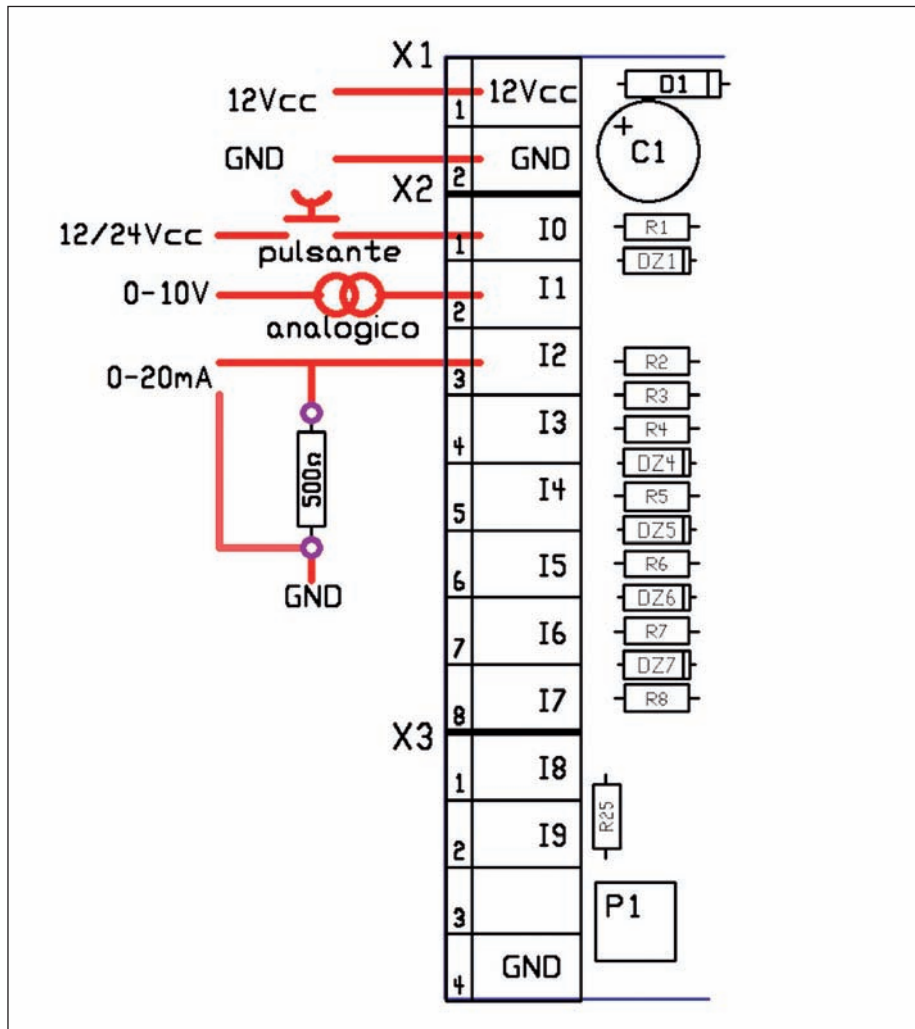



Figura 2 Schema semplificato del collegamento degli ingressi

sante P4. L'intero programma può essere cancellato premendo e tenendo premuto P3 dalla fase di programmazione fintanto che non appare la scritta

!DELETE PROGRAM!

SET=canc END=rtn

sicuramente a questo punto avete capito quali pulsante vanno premuti per la carcerazione completa o per il ritorno lasciando inalterato il programma.

La logica di funzionamento applicata dal PLC a ciascuna stringa è del tutto uguale a quella del vecchio progetto pubblicato qualche anno fa in queste stesse pagine. A parte i timer, la configurazione dei contatori e le operazioni con le analogiche, essa è piuttosto rigida e prevede sempre l'operazione tra due operandi e un operando di destinazione.

Per prima cosa esaminiamo il significato dei vari simboli da utilizzare durante la programmazione delle stringhe.

Operandi:

I ingresso digitale: indirizzo da 0 a f per un totale di 16 ingressi

Q uscita: indirizzo da 0 a f per un totale di 16 uscite

M marker: indirizzo da 0 a f per un totale di 16 marker

T timer: indirizzo da 0 a 7 per un totale di 8 timer da 1 a 250 sec/min

C contatore: indirizzo da 0 a 3 per un totale di 4 contatori da 1 a 65000 e peso da 1 a 100

A ingresso analogico: indirizzo da 0 a 3 per un totale di 4 ingressi normalizzati 0-10V

P configuratore del contatore: configura il contatore con il valore da raggiungere e lo attiva

Operazione tra bit:

& logica booleana AND tra due bit

| logica booleana OR tra due bit

! NOT: lo stato del bit viene invertito prima dell'operazione

T richiesta di configurare un Timer

Operazione uscita:

= il risultato logico è applicato all'uscita

S set (bit = 1)

R reset (bit = 0)

C complementa uscita (simula il relè Passo passo)

OPERAZIONE TRA BIT

E' la più comune tra le operazioni usate nel PLC e riguarda un confronto logico tra due bit siano essi ingressi, uscite, timer o marker, le troverete in qualsiasi occasione anche nei programmi dei PLC più evoluti. Iniziamo la nostra lezione con una semplice stringa che implementa tale logica e familiarizza con la scrittura di tipo "a lista di istruzioni"

00 I.0 & I.1 =Q0

da sinistra verso destra il primo numero rappresenta una posizione della stringa all'interno della memoria, **00** sarà la prima stringa elaborata ad ogni scansione. **I.0** è il primo operando ed è l'ingresso **I0**, il punto indica che la logica non è invertita, **I.1** è il secondo operando ed è l'ingresso **I1**, l'operazione logica richiesta è **AND** e il simbolo di uguale che precede **Q0** significa che all'uscita **Q0** viene applicato il risultato dell'operazione logica **AND** tra **I.0** e **I.1**. Possiamo scrivere la stessa riga di programma in maniera più chiara

Q0 = I0 AND I1

Dalla teoria alla pratica in poche mosse, inseriamo una stringa nel PLC, confermiamo quando richiesto e rendiamo il PLC operativo premendo il pulsante P4. Dopo aver applicato un LED all'uscita **Q0**, proviamo ad applicare all'ingresso **I0** e **I1** una tensione tra i 12 e i 24 Volt. È facile notare che solo quando i due ingressi si trovano allo stato logico 1 il LED in uscita si accende. Infatti se consultiamo una qualsiasi tabella booleana in logica AND non possiamo che leggere lo stesso risultato.

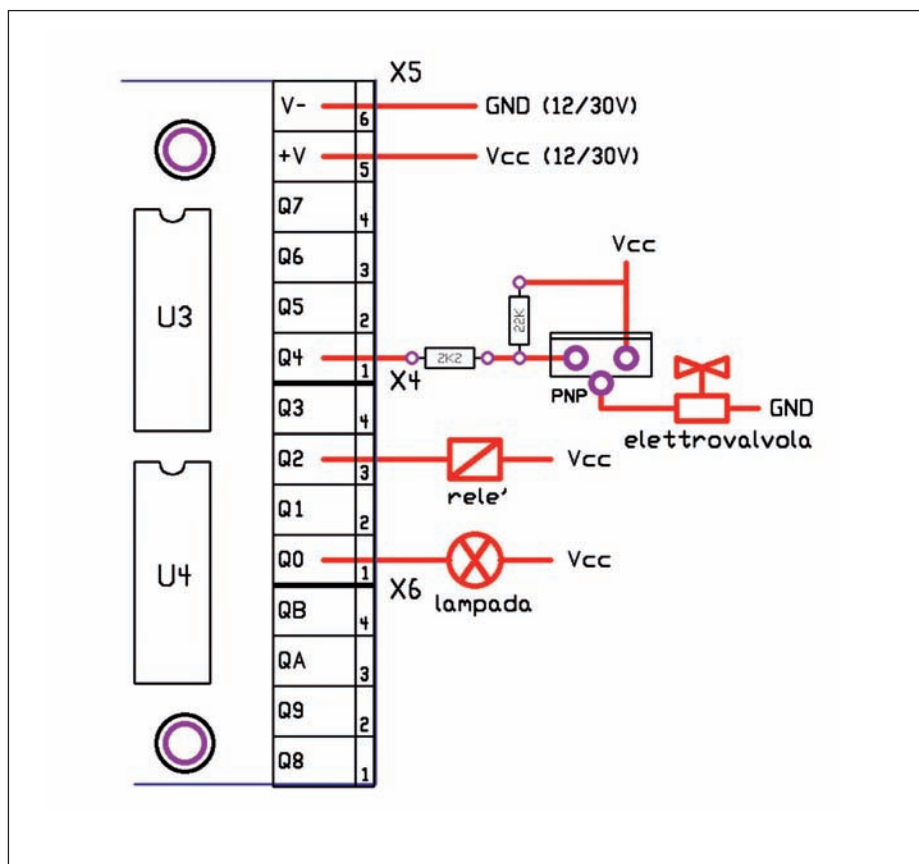



Figura 3: Figura 3 schema semplificato del collegamento delle uscite

Nota: durante il test e il collaudo è possibile evitare di collegare qualsiasi carico alle uscite e ricorrere piuttosto alla visualizzazione della pagina dello stato degli ingressi e delle uscite direttamente sul LCD, oppure collegare il PLC al PC con un cavo USB ed avviare il programma "*MiniPLC_V10.exe*". Del programma in Visual Basic ne parleremo il prossimo mese.

Ritorniamo in programmazione e modifichiamo la riga #**00** come segue

00 I.0 | I.1 =Q0

e procediamo con l'analisi pratica. Questa volta abbiamo posto in relazione i due ingressi con la logica **OR** e significa che qualsiasi ingresso posto ad 1 logico attiva l'uscita.

Abbiamo visto com'è facile eseguire un operazione logica tra due bit, ma se l'applicazione richiede tre o più ingressi in logica booleana, come risolvere il problema? Per la soluzione facciamo affidamento ai marker. Immaginiamo di avere la necessità di attivare l'uscita **Q0** solo quando gli ingressi **I0**, **I1** e **I2** sono posti a 1 logico digitando le seguenti stringhe

00 I.0 & I.1 =M0

01 M.0 & I.2 =Q0

nella prima stringa il risultato logico dell'operazione è affidato ad un marker, nella seconda operazione è tra il marker ed il terzo ingresso mentre il risultato viene finalmente applicato ad una uscita fisica. In pratica il Marker, definito come relè virtuale, è una particolare area di memoria del PLC in cui trattenere lo stato logico di alcune operazioni senza la necessità di variare la stato delle uscite.

Sovente capita di dovere implementare dei circuiti in cui possono coesistere tipi diverse di operazioni, vediamone un esempio

00 I.0 & I.1 =M0

01 M.0 | I.2 =Q0

modifichiamo le due stringhe precedenti sostituendo semplicemente l'operazione booleana **AND** della seconda riga con un **OR** e proviamo a sperimentare. L'uscita **Q0** sarà attiva quando gli ingressi **I0** e **I1** assieme oppure il solo l'ingresso **I2** sarà posto a 1 logico.

Nei nostri esempi abbiamo fatto uso di ingressi e marker come operandi, allo stesso modo possiamo usare uscite, timer, contatori. Di timer e contatori ne parleremo più avanti, l'applicazione delle uscite è piuttosto intuitivo, provate a modificare le due stringhe e testatele sul PLC.

00 I.0 & I.1 =Q0

01 Q.0 & I.2 =Q1

La logica invertita **NOT** è un'altra funzione logica molto importante. E il suo funzionamento è presto detto, al bit associato a tale funzione, viene invertito il suo stato logico prima di compiere una qualsiasi operazione. Modifichiamo la prima delle due righe finora memorizzate come segue e proviamo a testare funzionamento sul PLC.

00 I!0 & I.1 =Q0

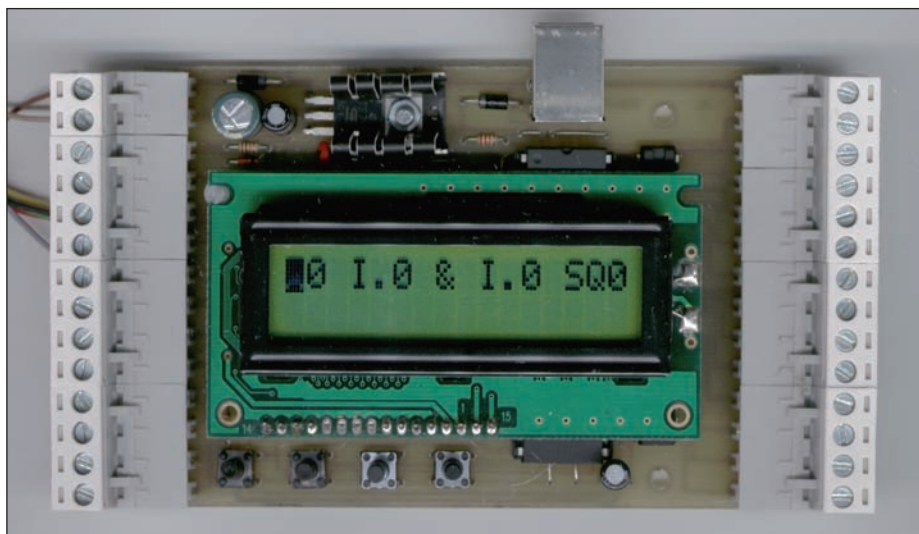
Potete notare che il funzionamento logico dell'ingresso **I0** è invertito. La prima applicazione che ci viene in mente di utilizzo della funzione **NOT** è quella del pulsante di stop del circuito di avviamento di un motore elettrico. Per questioni di sicurezza il pulsante di stop deve possedere i contatti di tipo NC (normalmente chiuso) ed è l'ideale per la programmazione con logica invertita.

Finora abbiamo visto come trasferire lo stato logico di un'operazione in un'uscita, ma se abbiamo bisogno di mantenere lo stato in uscita indipendentemente dalla variazione dello stato logico in ingresso dobbiamo utilizzare un'altra funzione. La prima di queste che andremo ad analizzare è il SET (**S**).

Abbiamo parlato di avviamento motore elettrico vediamo assieme come implementarlo con poche righe di programma. Iniziamo con la stringa

00 I.0 & I.0 SQ0

immaginiamo che all'ingresso **I0** sia collegato il pulsante di marcia e all'uscita **Q0** al contattore che alimenta il motore elettrico. La pressione del pulsante applicato all'ingresso **I0** attiva l'uscita **Q0** e di conseguenza viene alimentato il contattore del motore. Quando rilasciamo il pulsante, contrariamente agli esempi precedenti, l'uscita non viene disattivata ma resta attiva fintanto che non viene tolta l'alimentazione al circuito. Chiaramente in questa condizione diventa molto scomodo lavorare ed è improponibile presentare una soluzione del genere, è qui

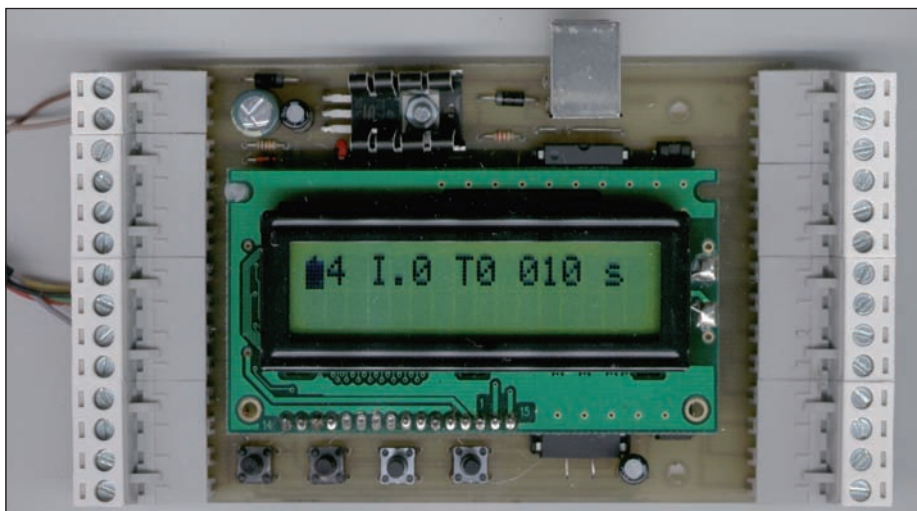
che ci viene in aiuto la funzione RESET (**R**).
Aggiungiamo la seconda stringa
01 I!2 & I!2 RQ0
In questo caso il pulsante di arresto di tipo NC viene collegato all'ingresso **I2**. Adesso il circuito funziona in maniera corretta, il pulsante di marcia avvia motore attraverso il contattore collegato all'uscita **Q0** e resta attivo fintanto che non viene perduto il pulsante di arresto. Possiamo complicarci la vita collegando un fine corsa di tipo NC ad un riparo della macchina elettrica che andiamo ad alimentare come circuito di sicurezza, ad esempio il coperchio delle cinghie per il cambio di velocità di un trapano a colonna, collegandolo all'ingresso **I1**. Le due semplici stringhe in grado di compiere le operazioni desiderate sono
00 I.0 & I.1 SQ0
01 I!2 I I!1 RQ0
nella prima stringa l'avviamento va a buon fine solo se il riparo è chiuso, mentre nella seconda, l'arresto della macchina è demandato sia a pulsante di stop che all'eventuale apertura del riparo.
Nota: essendo la logica di funzionamento del PLC ad analisi sequenziale delle istruzioni, è molto importante che la funzione SET preceda sempre la funzione RESET. Per una questione di sicurezza è di vitale importanza che il numero di stringa della funzione RESET sia sempre superiore a quella del SET.
Abbiamo lasciato come ultima analisi la gestione delle uscite come COMPLEMENTO (**C**). Possiamo associare il funzionamento di tale funzione al relè passo passo, se la funzione logica in ingresso è vera l'uscita viene complementata, ovvero se si trovano stato logico 1 passa a quello 0 e viceversa. Vediamo di chiarire il concetto con un semplice esempio
00 I.0 & I.0 CQ0
nota: se avete nella memoria del PLC l'esempio precedente quest'ultimo non può funzionare data la presenza del RESET all'uscita **Q0** della seconda riga di programma. Per poter procedere potete cancellare la riga #**01** o modificarla cambiando l'indirizzo dell'uscita.
Il funzionamento piuttosto semplice, ad ogni pressione del pulsante collegato all'ingresso **I0** l'uscita **Q0** cambio di stato.

TIMER

Un PLC senza timer serve a ben poco, il nostro ne ha ben otto. Ognuno di questi può essere caricato con un tempo limite fino a 250 secondi o minuti. La condizione logica in ingresso, se vera, determina sia il caricamento dei parametri che l'attivazione al conteggio del tempo. Se questa decadde prima che il timer abbia raggiunto il tempo richiesto, il timer viene disinstallato e il tempo fino a quel momento conteggiato, azzerato.
L'implementazione avviene con una semplice riga di programma
00 I.0 T0 005 s
in questo esempio ad essere attivato è il timer **T0** se, e solo se, la condizione logica di **I0** è vera, il tempo da raggiungere è di 5 secondi mentre la stringa
00 I.0 T0 005 m
indica che il tempo impostato è di 5 minuti. L'utilizzo dei timer nelle operazioni booleane avviene alla stregua di qualsiasi bit, gli esempi seguenti ne solo una prova.
01 I.0 & T.0 =Q0
In questo caso **T0** è confrontato con **I0** in logica AND. Potremo avere bisogno di lavorare solo con il bit del timer, provate le due stringhe successive tra loro equivalenti.

01 T.0 & T.0 =Q0
01 T.0 | T.0 =Q0
E' pure possibile fare lavorare i timer tra loro in sequenza. Testiamo un semplice programma in grado di avviare in successione tre utenze diverse.
00 I.0 T0 005 s
01 I.0 & T.0 =Q0
02 Q.0 T1 005 s
03 I.0 & T.1 =Q1
04 Q.1 T2 005 s
05 I.0 & T.2 =Q2
Tutto dipende dallo stato logico di **I0**, quando questo è vero inizia il conteggio del timer **T0**, una volta raggiunto il tempo impostato, attiva l'uscita **Q0**. L'uscita **Q0** abilita al conteggio il timer **T1** il quale controlla l'uscita **Q1**, e così via. Una volta terminato il ciclo di avvio le uscite restano attive fintanto che l'ingresso **I0** è alimentato. Quando la condizione logica di **I0** è falsa, i timer sono esclusi dal conteggio, i relativi contattori resettati e le uscite disabilitate. Da notare che il nostro PLC offre la possibilità di monitorare i timer in tempo reale. Premette il pulsante P1 finché visualizzate una delle pagina dei timer che si propone con questa grafica
T0:__0s T1:__0s
T2:__0s T3:__0<
mentre il pulsante P2 permette di passare da una pagina all'altra ovvero visualizzare alternativamente i timer da **T0** a **T3** e da **T4** a **T7**. La pagina sopra riportata rispecchia la situazione dell'esempio di avviamento sequenziale. Se attivato l'ingresso **I0**, i contatori dei timer incrementeranno il conteggio per poi fermarsi una volta raggiunto il valore impostato. Il simbolo "**<**" stampato subito dopo il valore del tempo indica che il timer non è richiamato da nessuna stringa in memoria e quindi rimane inutilizzato, mentre se indicato "**s**" o "**m**" significa che il timer è impostato da una stringa e indica il valore del tempo espresso rispettivamente in secondi o minuti.

CONTATORE

Una delle funzioni importanti aggiunte al nuovo PLC rispetto al vecchio modello è quella del contatore. È possibile implementare fino a quattro contatori il cui indirizzo va da **C0** a **C3**. È interessante notare che qualsiasi ingresso può essere adibito al conteggio, allo stesso modo di Marker e uscite. Inoltre è possibile visionare lo stato dei quattro contatori in un'unica pagina del LCD agendo sul pulsante P1 fino a visualizzare la pagina
C0:____0 C1:____0
C2:____0 C3:____0
essendo l'implementazione del contatore piuttosto complessa ed articolata rispetto per esempio a quella del timer, iniziamo la spiegazione con la sua configurazione e ciò avviene in un'unica stringa.
00 PC0 ___10 001
La stringa presa come esempio non fa altro che caricare nei registri, che potremmo chiamare "target" e "peso" del contatore **C0**, rispettivamente il valore da raggiungere e il valore da sommare al registro "target" ad ogni evento che influenzi il conteggio. Il primo registro può contenere un valore da 1 a 65000 mentre il secondo da 1 a 100.
Vediamo di integrare l'esempio precedente collegando il contatore **C0** con un ingresso qualsiasi programmando la seconda stringa
01 I.5 & I.5 =C0
e visualizzare a display l'avanzamento del contatore **C0** ogni volta che operiamo sull'ingresso **I5**
C0:____8 C1:____0
C2:____0 C3:____0
possiamo notare che il contatore **C0** avanza di un'unità perché abbiamo imposto il peso ad **1**. Modifichiamo la stringa **00** nel modo seguente
00 PC0 ___10 002
in questo caso il contatore **C0** avanza di **2** unità ogni volta che agiamo sull'ingresso **I5**. Se state seguendo queste pagine testando le stringhe in tempo reale, vi sarete sicuramente accorti che nonostante il contatore **C0** raggiunga il valore imposto questo continui imperterrito ad incrementare ogni volta che interveniamo sull'ingresso **I5**. Ciò è normale perché non abbiamo ancora associato al contatore nessuna uscita, non importa se fisica o virtuale. Provvediamo immediatamente con la stringa
02 C.0 & C.0 =Q8
che si occupa di attivare l'uscita **Q8** non appena il contatore **C0** ha raggiunto il valore prefissato. Questa volta possiamo agire più volte sull'ingresso **I5** senza ottenere nessun incremento oltre a quello impostato. È doveroso fare alcune considerazioni sul registro "peso", la prima riguarda il valore che non può essere superiore a quella del registro "target", nella seconda dobbiamo precisare che se il registro "target" viene programmato con un valore non divisibile con quello del registro "peso", il contatore agirà sull'uscita quando il valore raggiunto sarà ovviamente multiplo del valore del registro "peso" ma comunque uguale o superiore a quello del registro "target". Riprendiamo l'esempio precedente modificando la stringa
00 PC0 ___10 003
il registro "target" (10) non è divisibile per 3 e il contatore **C0** agirà sull'uscita quan-

do avrà raggiunto il valore 12. Quest'ultima considerazione è molto importante e dovete tenerne conto durante la configurazione del contatore per evitare eventuali errori di misura.
Allo scopo di chiarire l'argomento contatore, proseguiamo con un esempio pratico. Ipotizziamo di trovarci nella necessità di controllare il taglio di una trancia di una sbobinatrice di lamiera zincata. Immaginiamo di dover tagliare delle fette di lamiera lunghe 600 mm e di avere un riferimento di 25 millimetri ottenuto, per esempio, da una corona seghettata con dieci denti e applicata solidale ad un cilindro del diametro di 80 mm, sviluppo = 80*3,14 = 251mm/10denti = 25 mm circa. Un sensore induttivo applicato in prossimità dei denti della corona invia un impulso all'ingresso del PLC ogni 25 mm di avanzamento della lamiera. Vediamo quali sono le stringhe necessarie all'implementazione dell'intero programma e di seguito la spiegazione di ogni stringa.

00 I.0 & I.0 SQ0
01 I!1 & I!1 RQ0
02 PC0 __600 025
03 I.2 & I.2 =C0
04 C.0 & C.0 RQ0
05 C.0 & Q !0 SQ1
06 I.4 & I.4 RC0
07 Q.1 & I .4 RQ1

La prima stringa agisce con la funzione SET sull'uscita **Q0** la quale ecciterà un relè in grado di mettere in funzione intera macchina, mentre la seconda stringa testa lo stato del pulsante di arresto, necessario ad arrestare la macchina in caso di necessità o emergenza. La stringa #**02** parametrizza il contatore **C0** con la lunghezza limite di 600 mm ed un peso di 25 mm, in pratica ad ogni impulso del sensore induttivo **S1,** il contatore **C0** incrementa il proprio valore di 25 unità. La terza stringa controlla lo stato del sensore induttivo **S1** applicato all'ingresso **I2**, ricordando che quest'ultimo è applicato vicino ai denti della corona solidale ad uno dei rulli di trascinamento della lamiera e, nel nostro esempio, ogni lettura corrisponde ad un decimo di giro. Quando il contatore **C0** raggiungere o supera il valore di target interviene la stringa #**04**, la quale arresta la macchina diseccitando l'uscita **Q0**, successivamente la stringa #**05** fa scendere la lama per tranciare la lamiera. Dal momento che la trancia va a premere il fine corsa **FC2**, collegato all'ingresso **I4** del PLC, la stringa #**06** azzera il contatore **C0** e la stringa #**07** permette alla trancia di riposizionarsi nella condizione di riposo. Per attivare un altro ciclo, è necessario premere il pulsante di avvio collegato all'ingresso **I0** del PLC. Otto semplici stringhe e una scheda autocostruita per risolvere un'automazione tutt'altro che semplice, che potremmo completare con un contatore che tenga conto del numero di lamiere prodotte in questo modo:

08 PC0 __100 001
09 Q.1 & Q.1 =C1
10 I.6 & I.6 RC1

la stringa #**08** attiva il contatore **C1** in cui il target può assumere qualsiasi valore mentre il peso deve avere il valore 1. La #**09** incrementa il contatore **C1** di una unità ogni volta che interviene la trancia. E' possibile azzerare il contatore **C1** premendo un pulsante collegato all'ingresso **I6**. Il numero delle lamiere prodotte potrà essere visualizzato dalla comoda pagina principale nella modalità contatori (**C1**).

C0:__600 C1:___24
C2:____0 C3:____0

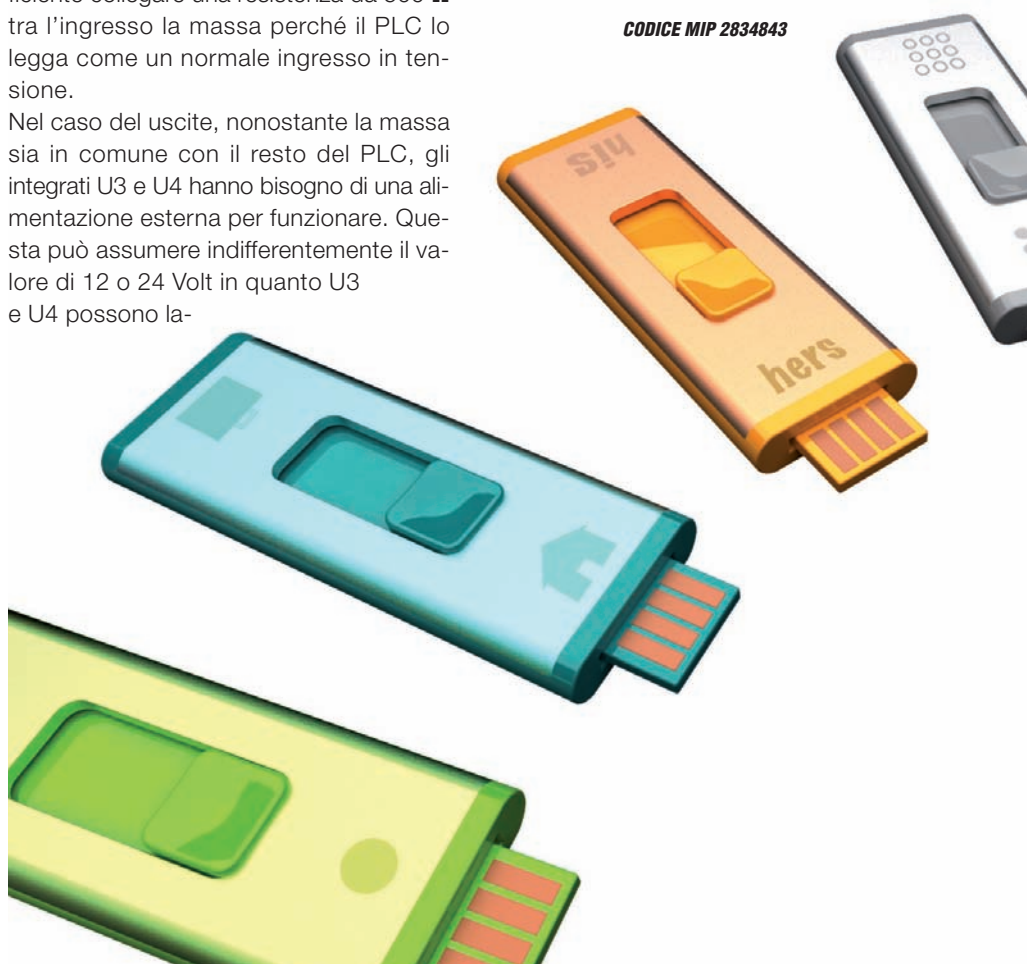
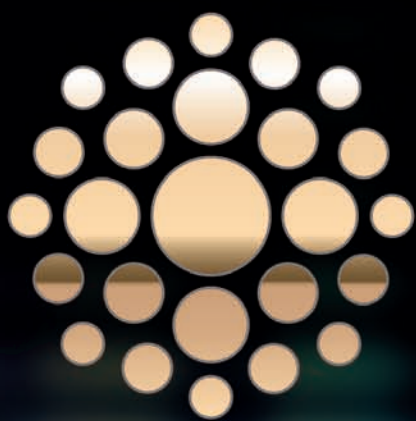
Ovviamente il programma proposto è solo un esempio di ciò che possiamo implementare con il nostro PLC, è possibile automatizzare completamente la macchina in modo da produrre un certo numero di lamiere o altre funzioni che l'operatore del settore avrà certamente in mente. Come abbiamo accennato all'inizio dell'articolo il nostro PLC esegue tutte le istruzioni disponibili in memoria e visualizza lo stato o il valore di alcuni parametri direttamente sul LCD, definendo questo insieme di azioni col nome di ciclo. Dal momento che un contatore ha rilevato lo stato alto del proprio ingresso, per poterlo rileggere nuovamente deve passare almeno un ciclo in cui questo si trovi allo stato logico basso. Il tempo minimo che può passare tra un impulso e l'altro applicato all'ingresso del contatore può quindi essere di due cicli, essendo però questi non sincronizzati con il sensore applicato all'ingresso, dobbiamo considerare come tempo minimo quello di tre cicli. Il che significa che devono passare circa 75 ms tra un impulso e l'altro, definendo in questo modo la frequenza massima di conteggio che risulta di 13 letture al secondo.

ANALOGICHE

Contrariamente ai contatori, gli ingressi analogici hanno una posizione fissa sulla morsettiera. Ad obbligarci ad una collocazione hardware rigida intervengono due fattori determinanti: il primo è la necessità di precedere l'ingresso fisico del microcontrollore da un buffer che ne adatti l'impendenza, il secondo riguarda la mappatura degli ingressi analogici del micro che sono prestabiliti in fabbrica. È utile notare che l'ingresso analogico richiamato da una stringa parametrizza i pri-

mi quattro ingressi da digitale ad analogico solo il tempo necessario alla lettura, dopodiché ritornano a lavorare come ingressi digitali.
La disposizione degli ingressi analogici è molto semplice
A0 corrisponde a **I0**
A1 corrisponde a **I1**
A2 corrisponde a **I2**
A3 corrisponde a **I3**
La funzione analogica dipende esclusivamente dal primo operando e si presenta nel seguente modo
00 A0 > 0,00 SQ0
A0 è l'ingresso analogico e, come anticipato, può assumere il valore tra **A0** e **A3**
Il simbolo "**>**" significa proprio "maggiore di" cioè se l'ingresso analogico è maggiore del valore seguente allora porta a termine l'azione in uscita. L'alternativa è rappresentata dal simbolo "**<**" e impone l'azione in uscita solo se l'ingresso analogico è inferiore al valore impostato.
00,0 è il valore da confrontare e accetta un numero tra **0,01** e **10,00** ovvero tra 0,01 e 10,00 Volt, anche se, per motivi pratici, il massimo valore impostabile è **9,99**.
S (SET) è l'azione in uscita mentre l'alternativa è **R** (RESET).
Q0 rappresenta il tipo di uscita, la quale, assieme al **M** (MARKER), può assumere qualsiasi indirizzo disponibile tra **0** e **f**.
Vediamo di chiarire i concetti con qualche esempio pratico
00 A0 > 5,00 SQ8
applicando una tensione all'ingresso **I0**, quando questa supera i 5 Volt viene settata l'uscita **Q8**. Se fate questa prova vi accorgere che una volta attivata l'uscita **Q8** quest'ultima non cambia di stato nonostante l'ingresso **I0** sia sottoposto ad un potenziale inferiore ai 5 Volt. Dobbiamo completare l'implementazione aggiungendo un ulteriore stringa
01 A0 < 4,90 RQ8
in questa abbiamo deciso di resettare l'uscita **Q8** quando l'ingresso analogico **A0** assume potenziale inferiore ai 4,90 Volt. In questo modo possiamo decidere il valore di isteresi da dare alla nostra applicazione.
Anche per questa funzione è possibile visualizzare il valore di ciascun ingresso analogico selezionando, durante la normale funzionamento del PLC, la pagina riservata alla funzione.
A0:_0,00 A1:_0,00
A2:_0,00 A3:_0,00
Ricordiamo che è possibile il collegamento a una fonte di corrente 0-20 mA semplicemente collegano una resistenza tra l'ingresso analogico e la massa. Tale resistenza deve avere il valore di 500 Ω, per esempio unendo in parallelo due resistenza da 1 KΩ 1/2W.

Nelle figure 2 e 3 vi proponiamo alcuni possibili collegamenti di sensori nella morsettiera degli ingressi e alcuni attuatori per le uscite. In riferimento alla figura 2, notiamo un pulsante collegato al morsetto 1 di X2, il pulsante può essere semplicemente sostituito dal contatto di un relè, l'uscita di un sensore induttivo o di una fotocellula, ottenendo lo stesso risultato. Ricordando che il nodo comune degli ingressi è rappresentato dal morsetto 2 di X1 (GND), possiamo collegare un dispositivo con l'uscita analogica 0-10 Volt tra il morsetto GND è uno dei quattro ingressi analogici disponibili. Nella figura 2 si fa uso dell'ingresso **I1** collegato al morsetto 2 di X2, mentre al morsetto 3 è collegato un dispositivo che presenta l'uscita in corrente. Come anticipato è sufficiente collegare una resistenza da 500 Ω tra l'ingresso la massa perché il PLC lo legga come un normale ingresso in tensione.
Nel caso del uscite, nonostante la massa sia in comune con il resto del PLC, gli integrati U3 e U4 hanno bisogno di una alimentazione esterna per funzionare. Questa può assumere indifferentemente il valore di 12 o 24 Volt in quanto U3 e U4 possono lavorare con una tensione massima di 30 Volt, dipende tutto dalle caratteristiche degli utilizzatori collegati alle uscite. In pratica se collegate dei carichi funzionanti a 24 Volt, dovete alimentare le uscite ai morsetti 5 e 6 di X5 con 24 Volt. La massima corrente prelevabile da ciascuna uscita è di 300 mA, ma è consigliabile restare ben al di sotto di tale valore soprattutto se le uscite attive sono molte, poiché la massima dissipazione di U3 e U4 non è molto elevata. Potrebbe rendersi necessario collegare un carico che ha bisogno di una corrente superiore a quella fornibile da U3 o U4, in tal caso è possibile interporre un semplice transistor di tipo PNP tra l'alimentazione e il carico e la base del transistor controllata dall'uscita del PLC tramite un paio di resistenze. È possibile notare l'esempio pratico alla figura 3, l'uscita **Q4** controlla l'elettrovalvola tramite il transistor in formato TO220. Nella prossima ed ultima puntata presenteremo una piccola scheda in grado di misurare la temperatura e riproporla in uscita normalizzata a 0-10 Volt e descriveremo l'interfaccia i Visual Basic su cui visualizzare tutti i parametri del PLC in un'unica pagina del PC. ❏

CODICE MIP 2834843

di ROBERTO D'AMICO (IW0GTF)

Tutorial PIC18 *(parte nona)*

GESTIRE *le* INTERRUZIONI

I programmi rappresentano un flusso d'istruzioni sequenziali che vengono eseguite dalla CPU, ma a volte si ha la necessità di interrompere questo normale flusso perseguire delle istruzioni diverse per rispondere ad un particolare evento.

In alcune lezioni precedenti ci sono statti degli accenni agli Interrupt, che rappresentano sostanzialmente il modo che un microcontrollore ha di gestire determinati eventi, che non possono essere sincronizzati in altro modo con il codice.
Prendiamo ad esempio la conversione AD, come abbiamo visto dopo aver avviato il processo, il nostro programma di esempio interrogava ripetutamente un flag per attendere la fine della conversione, questo però faceva si che il microcontrollore restasse bloccato fino alla conclusione dell'operazione. Usando invece gli interrupt, una volta avviata la conversione, il micro è subito pronto per dedicarsi ad altre cose, quando questa sarà terminata, scatterà automaticamente la chiamata alla funzione che gestisce gli eventi d'interrupt.

GLI INTERRUPT DEL PIC

I PIC oltre a gestire interrupt per le varie periferiche, permettono anche di impostare un livello di priorità su essi, ma naturalmente possono funzionare anche senza la gestione della priorità. A prima vista la gestione degli interrupt può sembrare complessa, visto che per la loro configurazione sono utilizzati dieci registri, ma va ricordato che oltre alla gestione di base questi registri sono utilizzati per l'impostazione delle interruzioni di tutte le periferiche del PIC. In generale per ogni interrupt di periferica esistono tre bit di configurazione:

- **Priority**, questo bit serve ad indicare se l'interruzione dovrà essere gestita in alta priorità (bit a 1) o in bassa priorità (bit a 0). Le interruzioni definite in altra priorità, possono interrompere l'esecuzione di un'interruzione a bassa priorità o metterla in coda se è in esecuzione la gestione di un interrupt ad alta priorità;
- **Enable**, questo bit serve ad abilitare (bit a 1) l'interruzione su una specifica periferica;
- **Flag**, questo tipo di bit indica invece quando un interrupt è avvenuto.

Quando avviene l'evento che scatena un interrupt, ad esempio, il termine della conversione AD, l'esecuzione del programma viene spostata ad una specifica locazione di memoria, riservata per la gestione degli interrupt, se si utilizza la gestione delle priorità tale locazione ha indirizzo 0x18 per le interruzioni a bassa priorità e 0x08 per quelle ad alta priorità. In queste locazioni viene memorizzato l'indirizzo di memoria cui il Program Counter deve saltare per eseguire il codice che si occuperà di gestire le interruzioni. Come avrete sicuramente capito, non esiste uno specifico indirizzo di memoria per ogni interrupt di periferica, ma tutti accedono alla stessa locazione, sarà quindi il codice che andremo a scrivere che dovrà occuparsi di capire quale interrupt deve essere gestito. Per fare questo si utilizzano i bit Flag, i quali vengono automaticamente impostati ad 1 quando avviene un'interruzione. Nella programmazione in C per il C18, come vedremo in seguito, vengono definite due funzioni per gestire le due priorità, all'interno delle quali tramite dei controlli sui Flag, tipicamente degli if, possiamo determinare quale periferica, ha generato l'evento ed eseguire il codice per gestire l'interruzione. Bisogna ricordarsi, dopo aver eseguito il codice per gestire l'interruzione, di

azzerare il relativo Flag dell'interrupt, quest'operazione com'è logico non è automatica e deve essere eseguita manualmente.

COME CONTROLLARE UN SERVO RC

I servo motori RC, sono dispositivi dotati dal loro interno di una riduzione meccanica e di una circuiteria elettronica per gestirne i movimenti in funzione del segnale che gli viene inviato per pilotarli. Il loro connettore presenta due collegamenti per l'alimentazione e un terzo per il segnale necessario a pilotarli, che ricorda il PWM che abbiamo visto in una delle precedenti lezioni su Fare Elettronica, la frequenza di funzionamento è di 50Hz, che equivale a dire un periodo di 20mS, l'impulso invece ha un'ampiezza che può variare da 1mS a 2mS (in alcuni modelli questi valori possono essere leggermente diversi), questa variazione stabilisce di quanti gradi deve ruotare il servo motore, l'immagine di Figura 1 mostra la corrispondenza tra posizione del servo e frequenza del segnale.

IL CIRCUITO

Il circuito di questa lezione è semplicissimo, continuiamo ad utilizzare il circuito base di queste lezioni, al quale colleghiamo in più il servo motore. Questo ha un connettore a tre pin, normalmente il terminale Rosso corrisponde all'alimentazione positiva, quello Nero al collegamento verso massa, il Bianco (in alcuni modelli può essere giallo)

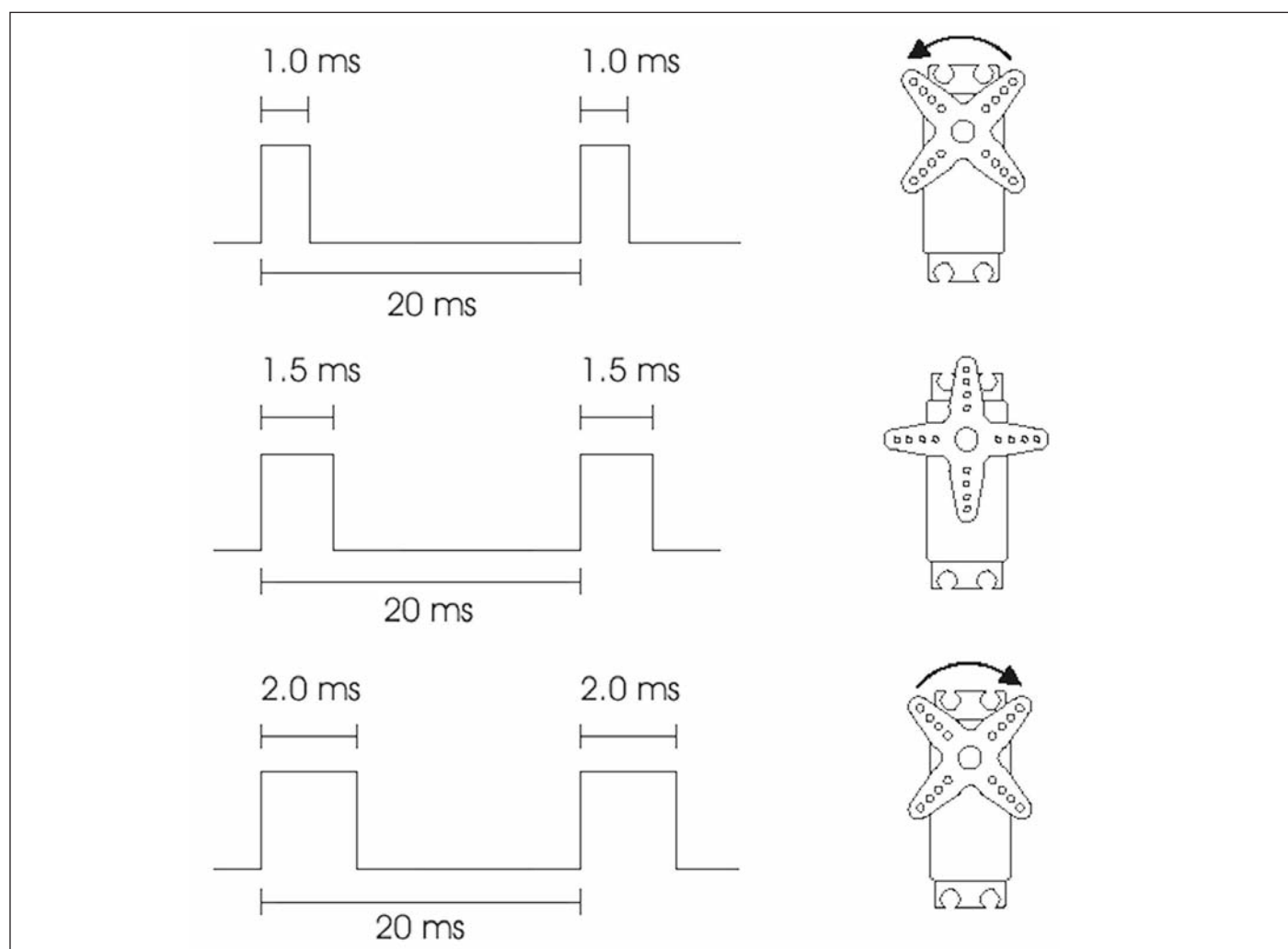



Figura 1: Posizione servo in funzione del segnale

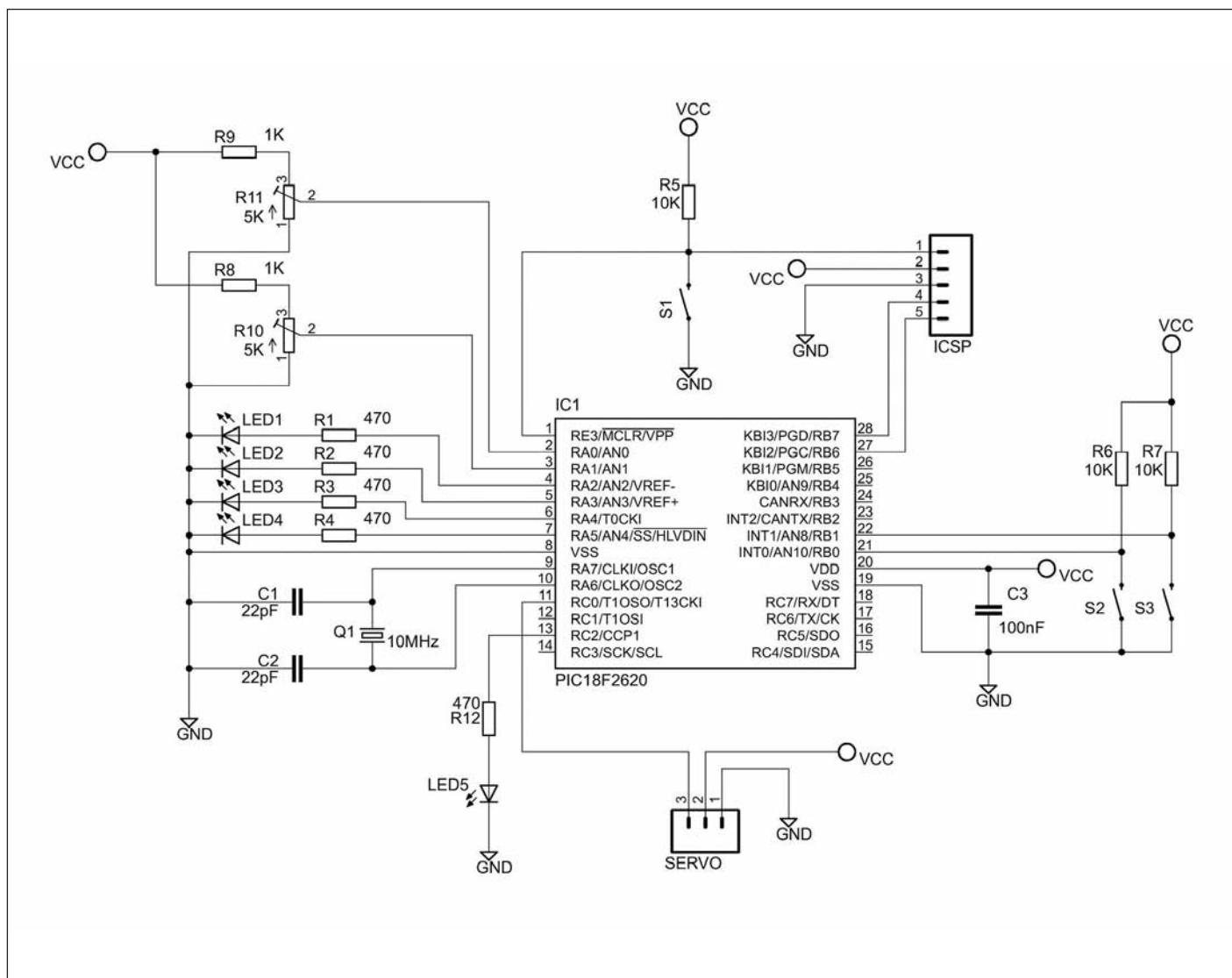



Figura 2: Lo schema elettrico del circuito

è il pin cui applicare il segnale di controllo, per il programma che stiamo per andare ad analizzare deve essere collegato al pin RC0 del PIC.

IL TIMER0

Tutti i modelli di PIC annoverano tra le loro periferiche uno o più Timer, quello che vediamo in questa lezione è il Timer0, che può funzionare a 8 o 16bit. Un Timer non è altro che un contatore che, può essere incrementato da un evento esterno al PIC, tramite un opportuno pin di ingresso, oppure tramite il clock interno, che può essere ridotto tramite la configurazione del prescaler. Nel registro TMR0 viene caricato il valore da confrontare con il contatore del Timer0, quando questi due valori sono uguali viene scaturito l'interrupt, questo permette di avere un'interruzione dopo un preciso intervallo di tempo. Il Timer0 si configura tramite i bit del registro T0CON, dove troviamo il bit T08BIT con cui viene impostato la dimensione del registro contatore (8 o 16 bit), T0CS permette invece di stabilire la fonte del clock per l'incremento del contatore, impostandolo a 0 utilizza il clock interno, che può essere a sua volta suddiviso tramite la configurazione dei bit che controllano la divisione di frequenza del prescaler che, a sua volta va abilitato tramite il bit PSA.
Per stabilire la durata del Timer0 possiamo utilizzare la seguente formula

```
Durata = TMR0*(4/Fosc)*PS
```

Dove TMR0 è il valore di comparazione del registro contatore, Fosc è la frequenza del clock del PIC e PS il valore di divisione del prescaler.
Solitamente però, quello che interessa calcolare è il valore da impostare in TMR0, perché gli altri dati sono quelli fisici del circuito o quelli che desideriamo impostare per i nostri scopi come la durata del timer, quindi viene utilizzata la seguente formula:

```
TMR0 = 256 – [(Durata * Fosc)/(4 *
PS)]
```

Supponendo di volere una durata di 1 microsecondo, di avere un quarzo da 40MHz ed il prescaler impostato a 64 abbiamo

LISTATO 1

```
#include <p18f2620.h>
#include <delays.h>
/**********
* DEFINE *
**********/
#define LED1 LATAbits.LATA2
#define LED2 LATAbits.LATA3
#define LED3 LATAbits.LATA4
#define LED4 LATAbits.LATA5
#define SERVO   LATCbits.LATC0
#define PULSANTE1 PORTBbits.RB0
#define PULSANTE2 PORTBbits.RB1
#define TMR0_VALUE      240
#define PERIODO_SERVO   200
/**************************
* Dichiarazione Funzioni *
**************************/
void InterruptVectorLow(void);
void InterruptHandlerLow(void);
void InterruptVectorHigh(void);
void InterruptHandlerHigh(void);
/*********************
* Variabili globali *
*********************/
unsigned char counter_servo=0;
unsigned char posizione_servo=15;
/********
* MAIN *
********/
void main(void)
{
   unsigned char counter=0;

   /*Reset iniziale delle porte*/
   PORTA=0;
   PORTB=0;
   PORTC=0;
   /*Definizione della funzione delle sin-
gole porte*/
   TRISA=0x03;
   TRISB=0b00000011;
   TRISC=0x00;
   /*Impostazione dei registri di configu-
razione del PIC per abilitare ADC sui pin
RA0 e RA1*/
   ADCON1=0b00001101;
   ADCON2=0b10000010;
   ADCON0=0b00000001;
   CMCON=0b00000111;
   /*Impostazione Timer0*/
   T0CONbits.T08BIT=1;
   T0CONbits.T0CS=0;
   T0CONbits.T0SE=0;
   T0CONbits.PSA=0;
   T0CONbits.T0PS0=1;
   T0CONbits.T0PS1=0;
   T0CONbits.T0PS2=1;
   T0CONbits.TMR0ON=0;
   /*Impostazione Interrupt*/
   RCONbits.IPEN=1;
   INTCONbits.GIEH=0;
   INTCONbits.GIEL=0;
   INTCONbits.TMR0IF=0;
    PIR1bits.ADIF=0;
   INTCON2bits.TMR0IP=1;
   IPR1bits.ADIP=0;
   INTCONbits.TMR0IE=1;
   PIE1bits.ADIE=1;
   INTCONbits.GIEH=1;
   INTCONbits.GIEL=1;
   /*Imposta il valore per il Timer0 e lo
abilita*/
   TMR0L=TMR0_VALUE;
   T0CONbits.TMR0ON=1;
   /*Accende il LED4 per indicare che il
programma è avviato*/
   LED1=1;
   LED2=1;
   LED3=1;
   LED4=1;
   Delay10KTCYx(250);
   LED1=0;
   LED2=0;
   LED3=0;
   LED4=0;
   /*Accende il primo LED*/
   LED1=1;
   Delay10KTCYx(250);
   // Loop infinito
   while(1)
   {
       /*Cicla per eseguire lo shift a sin-
istra del bit/LED*/
       for(counter=0;counter<3;counter++)
       {
           LATA=LATA<<1;
           Delay10KTCYx(250);
       }
       /*Cicla per eseguire lo shift a de-
stra del bit/LED*/
       for(counter=0;counter<3;counter++)
       {
           LATA=LATA>>1;
           Delay10KTCYx(250);
       }
   }
}
/***********************
* Low Interrupt Vector *
***********************/
#pragma code InterruptVectorLow = 0x18
void InterruptVectorLow(void)
{
   _asm
           goto InterruptHandlerLow
   _endasm
}
#pragma code
#pragma interruptlow InterruptHandlerLow
void InterruptHandlerLow(void)
```

Segue

LISTATO 1

```
{
   if(PIR1bits.ADIF==1)
   {
       /*Calcola la uova posizione del ser-
vo*/
       posizione_servo=(ADRES/100)+10;
       /*Azzera il flag dell'interrupt*/
       PIR1bits.ADIF=0;
   }
}
/*************************
* High Interrupt Vector *
*************************/
#pragma code InterruptVectorHigh = 0x08
void InterruptVectorHigh(void)
{
   _asm
           goto InterruptHandlerHigh
   _endasm
}
#pragma code
#pragma interrupt InterruptHandlerHigh
void InterruptHandlerHigh(void)
{
   if(INTCONbits.TMR0IF==1)
   {
       counter_servo++;
       /*Controlla se è passato il tempo
della fine dell'impulso*/
       if(counter_servo==posizione_servo)
       {
           /*Resetta l'uscita che pilota il
SERVO*/
           SERVO=0;
       }
       /*Se sono passati 20mS*/
       if(counter_servo==PERIODO_SERVO)
       {
           /*Azzera il contatore*/
           counter_servo=0;
           /*Setta l'uscita che pilota il
SERVO1*/
           SERVO=1;
           /*Avvia la lettura del trimmer*/
           ADCON0bits.GO=1;
       }
       /*Reimposta il valore del Timer0 ed
il Flag dell'interrupt*/
       TMR0L=TMR0_VALUE;
       INTCONbits.TMR0IF=0;
   }
}
```

256 – [(0,0001 * 40000000)/(4 * 64)]=240,375

Quindi utilizzando la parte intera, di questo valore calcolato, per valorizzare il registro TMR0, ci permette di avere un interruzione ogni millisecondo.

Figura 3: Il circuito montato sulla bread-board

MUOVIAMO IL SERVO

Il programma che andiamo ad analizzare in questa lezione prende spunto da listati di alcune lezioni precedenti, avremo la rotazione dei diodi Led tramite delle semplici istruzioni inserite all'interno della funzione main, come visto nella **Lezione 2**, questo servirà per vedere che il programma scritto all'interno della funzione principale continua a girare correttamente, anche se avvengono delle interruzioni. In realtà il codice che è eseguito nella funzione main viene comunque interrotto, ma utilizzando delle routine di gestione degli interrupt (ISR - Interrupt Service Routine) molto snelle, quindi con tempi di esecuzione molto rapidi, il programma principale non risente di rallentamenti apprezzabili. Nel programma viene utilizzata la conversione AD, ma rispetto alla **Lezione 4**, questa volta il controllo di fine conversione è affidato ad un interrupt a bassa priorità. In questo caso il valore di tensione misurato sul trimmer viene utilizzato per definire la posizione del servo. In fine utilizziamo il Timer0, visto nel paragrafo precedente, per gestire le temporizzazioni necessarie a fa muovere il servo. Come abbiamo visto la frequenza del segnale per pilotare il servo è molto bassa e non può essere generata tramite l'apposito modulo per la ge-

nerazione di segnali PWM come visto nella **Lezione 5**, quindi utilizziamo l'interrupt sul Timer0 impostandolo per scattare ogni 100 micro secondi. Su YouTube è presente un filmato che mostra il programma in azione http://youtu.be/MvyHL-ENgb3w.

Analizziamo nel dettaglio il codice del **Listato 1**, tra le define le novità sono SERVO che indica LATC0 cui è appunto collegato il servo, TMR0_VALUE che indica il valore cui impostare il Timer0 per ottenere un'interruzione ogni 100 microsecondi ed infine PERIODO_SERVO utilizzata per definire dopo quante interruzioni del Timer si completa il periodo del segnale con cui è pilotato il servo che, come abbiamo visto nel paragrafo che ne spiegava il funzionamento, richiede un segnale con un periodo di 20 millisecondi, di conseguenza moltiplicando i 100 microsecondi di un'interruzione per 200, valore definito in PERIODO_SERVO otteniamo 20000 microsecondi che corrispondono a 20 millisecondi.
A seguire troviamo le dichiarazioni delle funzioni per la gestione delle interruzioni, per entrambe le priorità abbiamo due funzioni InterruptVector e InterruptHandler, il corpo di queste funzioni si trova nella parte finale del **Listato 1**, il primo tipo si occupa di indicare la funzione che gestisce le interruzioni, come detto all'inizio di questa lezione, definisce quindi il "vettore" alla vera funzione di gestione. Il secondo tipo di funzioni è invece quello che include il codice per controllare quale tipo di interrupt deve essere gestito, e le istruzioni necessarie a gestirlo, come vedremo più avanti.
Continuando a leggere il codice troviamo la dichiarazione di due variabili counter_servo e posizione_servo, dichiarandole all'esterno della funzione main sono definite di tipo globale cioè accessibili da qualsiasi funzione anche esterna alla main stessa. Questo tipo di definizione ci permette di condividere le informazioni di queste variabili tra le varie funzioni del programma, quindi funzione main e le due funzioni di gestione degli interrupt.
All'interno del main troviamo al solito la configurazione delle porte e delle periferiche, di nuovo abbiamo quella riguardante il Timer0, che viene configurato per funzionare con il registro ad 8 bit ed il prescaler 1:64. Il Timer rimane però disattivato, verrà attivato successivamente dopo la configurazione degli interrupt.
L'impostazione degli interrupt non è particolarmente complessa, è impostato il flag IPEN per abilitare la priorità sugli interrupt e subito dopo sono disabilitate le interruzioni di entrambi i tipi e azzerati i Flag degli interrupt che devono essere gestiti, nel caso del programma del listato, i Flag da gestire sono TMR0IF e ADIF. Con i bit TMR0IP e ADIP è impostata la priorità, come avrete sicuramente notato i bit di configurazione dell'interrupt riguardante il Timer0 fanno parte del registro INTCON e INTCON2, mentre quelli del convertitore AD sono tutti negli appositi registri delle interruzioni di periferiche. Per concludere sono abilitati gli interrupt del Timer0 e dell'ADC e in seguito le due priorità che erano state disabilitate in precedenza.

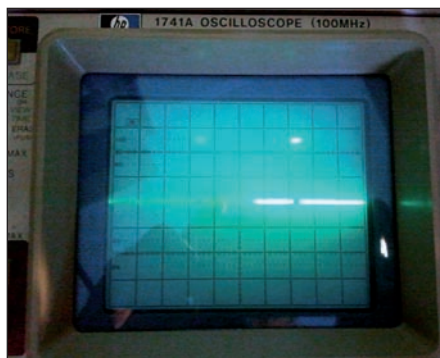



Figura 4 - Il segnale generato visto all'oscilloscopio.

L'ultima parte di configurazione riguarda il valore da impostare al Timer0 per determinare la sua durata e l'abilitazione per il conteggio.
Il programma continua con il lampeggio dei quattro led per poi passare al loop infinito, che al suo interno propone lo stesso codice che abbiamo visto nella seconda lezione di questo tutorial.
Vediamo ora nel dettaglio le funzioni che gestiscono gli interrupt, analizzando per prima la InterruptHandlerLow, che si occupa di gestire la fine della conversione AD, che abbiamo impostato come priorità bassa, questa funzione è chiamata ogni volta che scatta una qualsiasi interruzione di bassa priorità, quindi per prima cosa andiamo a verificare se il Flag ADIF è impostato ad 1, per poi recuperare il risultato della conversione dal registro ADRES e calcolare il nuovo valore da memorizzare nella variabile globale posizione_servo. L'ultimo passo è di azzerare il Flag ADIF, per permettere all'interrupt di essere gestito nuovamente.
Nella funzione InterruptHandlerHigh, dopo il controllo sul Flag TMR0IF, si trova il codice che si occupa di generare la forma d'onda necessaria a muovere il servo. La prima istruzione serve ad incrementare la variabile counter_servo, che si occupa di tenere il conteggio delle volte che avviene l'interrupt e quindi stabilire i tempi per il periodo e il semi periodo. Quando counter_servo raggiunge lo stesso valore della variabile posizione_servo, che abbiamo visto, è valorizzata dalla conversione AD, vuol dire che è terminato il primo semiperiodo e il segnale di uscita va posto a livello logico 0. Quando invece la variabile counter_servo, raggiunge il valore massimo definito in PERIODO_SERVO, vuol dire che è passato il tempo di un periodo (200mS), viene quindi mandata l'uscita a livello alto, azzerato il contatore e avviata una lettura dell'ADC. La gestione dell'interruzione si termina con il caricamento nel registro TMR0L, che indica il tempo del Timer0 e l'azzeramento del Flag dell'interrupt relativo.

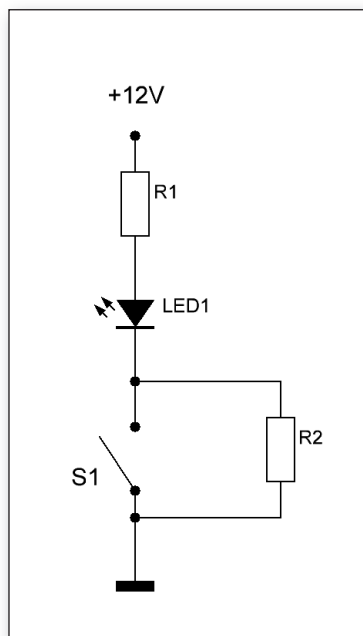
CONCLUSIONI

Abbiamo visto in questa lezione un argomento molto importate nel mondo dei microcontrollori, senza l'utilizzo degli Interrupt il PIC non avrebbe mai potuto eseguire le operazioni presenti nel listato **Listato 1**, o meglio avremmo dovuto scrivere del codice in grado di sincronizzare tutte le varie azioni (pilotare servo, leggere il valore del trimmer, scorrimento dei led) aumentando notevolmente la complessità del programma, con possibili problemi sempre dietro l'angolo.
Al solito vi suggerisco di giocare con il codice del Listato 1, per apprendere meglio la teoria, potreste ad esempio, con il supporto del DataSheet, aggiungere l'interrupt sulla porta seriale e controllare la posizione del servo tramite comandi inviati dal PC. ❑

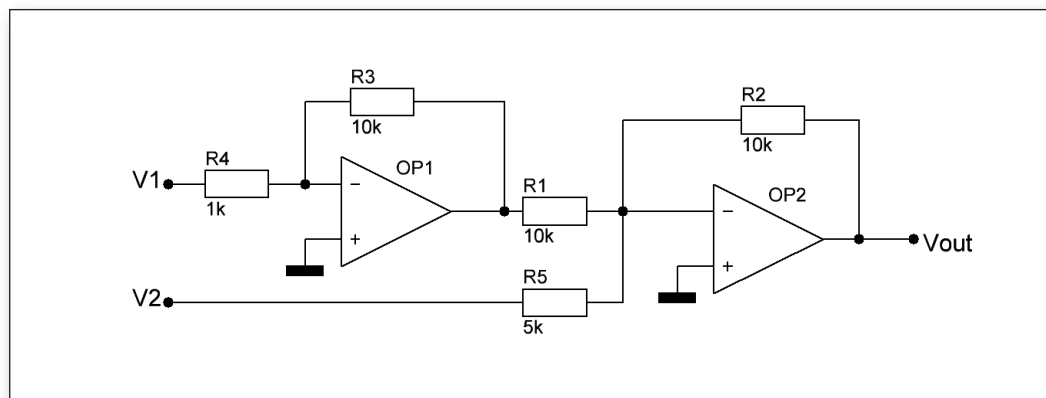
CODICE MIP 2821451

elettro quiz

332/333

Le risposte ai quiz "facile" e "difficile" vanno inviate esclusivamente compilando il modulo su www.farelettronica.com/eq specificando la parola chiave ***"Mylar"****.*
Le risposte e i vincitori (previa autorizzazione) sono pubblicati alla pagina www.farelettronica.com/eq a partire dal 15 del mese successivo alla pubblicazione sulla rivista.
A tutti i partecipanti verrà assegnato un buono sconto del 10% (validità 3 mesi dalla data di assegnazione) utilizzabile per un prossimo acquisto su www.ieshop.it

Premio:
Abbonamento al club di Fare Elettronica

base

Dimensionare i valori di R1 e R2 in modo che con S! aperto il LED sia percorso da una corrente di 7mA e con S1 chiuso la corrente salga a 10mA. Per i calcoli si consideri di 1,2V la caduta di tensione sul LED.

Premio:
Abbonamento a FE

avanzato

Considerando ideali i due amplificatori operazionali, determinare l'operazione artimetica su V1 e V2 eseguita dal circuito di figura.

di MASSIMILIANO MIOCCHI

INDUTTANZIMETRO *da* LABORATORIO

L'induttanzimetro è uno strumento molto utile in laboratorio. Ecco come costruirne uno molto semplice ma allo stesso tempo molto efficiente.

Da sempre per l'hobbista e non, lo strumento più prediletto dopo l'Oscilloscopio è l'induttanzimetro, capace di misurare il valore di un induttanza che spesso affligge con la sua presenza nei circuiti anche il progettista piu esperto. Spesso quando ci si accinge ad ingegnerizzare un prodotto, che esso sia un alimentatore o una centralina o un dispositivo oscillante o un amplificatore audio di certo è fondamentale l'uso delle induttanze e come accade è raro riuscire a realizzarle senza l'uso di questo fondamentale strumento di misura. Per poterlo realizzare ho dovuto dividere il progetto in due parti , la prima prevedeva la realizzazione di uno stadio oscillatore la cui frequenza variasse in funzione dell'induttore applicato e questo ha comportato non pochi problemi, la seconda ed ultima parte, era realizzare un frequenzimetro capace di leggere tale frequenza ed eseguire i calcoli opportuni per determinare il valore dell'induttore posto in misura. Misurare il valore di un induttanza non è complicatissimo sicuramente prevede degli accorgimenti che non sempre trovano riscontro nella semplicità e nella economicità dello stesso circuito. Uno dei metodi più utilizzati consiste nel leggere la variazione di frequenza in accordo con il circuito oscillante LC, un secondo metodo è il ponte

di Maxwell da cui si ottiene una misura precisa senza che intervengano problemi dovuti alle oscillazioni e quindi al prevalere di fenomeni di disturbo, come la temperatura o azioni esterne. Questo secondo metodo è una misura diretta con la quale è possibile conoscere anche il valore della resistenza offerta dall'induttore e con diversi accorgimenti è altresì possibile determinare il fattore Q definito fattore di merito, inteso come rapporto Q = Xl/Rs. Tanto più piccola è la resistenza statica dell'induttore tanto maggiore sarà il suo fattore di merito che rende l'induttore di buona qualità. Un tipico schema riassuntivo del ponte di Maxwell è quello raffigurato in figura a - b - c. Come noterete abbiamo la figura a che rappresenta il ponte di Maxwell, la b è la sua applicazione completa delle compensazioni e la c in definitiva rappresenta lo schema a blocchi di come concettualmente dovrebbe essere ingegnerizzato il tutto. E' decisamente più complesso......... è chiaro che la misura indiretta che ho scelto per ovvi motivi di convenienza e semplicità ha prevalso, dando comunque vita ad un ottimo strumento di alta precisione.

Figura1: schema elettrico della sezione di controllo e gestione dell'apparato.

SCHEMA ELETTRICO

Avendo optato per la realizzazione dell'induttanzimetro partendo da un concetto di misura indiretta, ho diviso il circuito in due sezioni, la figura 1 riguarda la sezione di controllo e di gestione dell'apparato, la figura 2 è il cuore del sistema; per i più intraprendenti e un classico circuito di oscillatore ma con una peculiarità, e realizzato con un comparatore LM311 insostituibile per questa applicazione date le sue caratteristiche, anche se è comunque utilizzabile in sua sostituzione il Max4413 di non facile reperibilità. La figura 1 è la prima sezione ad essere stata realizzata, sostanzialmente è un frequenzimetro dotato di prescaler. Il microcontrollore utilizzato per questa applicazione è un Atmega328 lo stesso montato da Arduino, vedi IC1; la sua funzione è quella di leggere la frequenza presente in ingresso al pin 14 dopo essere stata scalata dal prescaler IC6 un comunissimo contatore a dodici stadi CD4020. Come vedremo in seguito quando parleremo dell'oscillatore la frequenza a cui ci riferiamo è dell'ordine dei centinaia di KHz, data la frequenza cosi alta ho ritenuto necessario scalarla di 2048 volte per assicurarmi la massima precisione possibile nella lettura. La scelta sull'utilizzo del prescaler è dettata da diversi aspetti: Il primo dovuto all'impossibilita di agganciare il segnale da leggere a frequenze cosi alte senza perdere precisione, la seconda dovuto ad un eccessivo uso della CPU alle richieste di interrupt indi-

Figura 2: schema elettrico del cuore dell'induttanzimetro

spensabili per il conteggio; diversamente, non utilizzare gli interrupt, comporterebbe perdita di informazioni e quindi precisione. Ecco perchè un prescaler è la giusta soluzione. Gli integrati IC2A e IC2B sono dei comunissimi SN7404, due dei quali sono utilizzati per visualizzare i due stati di IC1, il primo led rosso quando è in fase di calibrazione, il secondo led verde a calibrazione avvenuta e in attesa di effettuare la misura, IC2C e IC2D in parallelo bufferizzano l'uscita dell'oscillatore con il prescaler migliorandone l'accoppiamento e il segnale. Il transistor T1 ha il compito di pilotare un relay di tipo reed (silenzioso) il cui compito è quello di garantire la messa a massa dell'induttanza campione L1 (induttanza di precisione da 100uH) per effettuare la calibrazione dello strumento, terminata la calibrazione T1 si interdice e il relè si stacca isolando dalla massa l'induttore e predisponendo cosi l'apparato ad effettuare una prima misura. Ricordo ai lettori che tutto questo è gestito

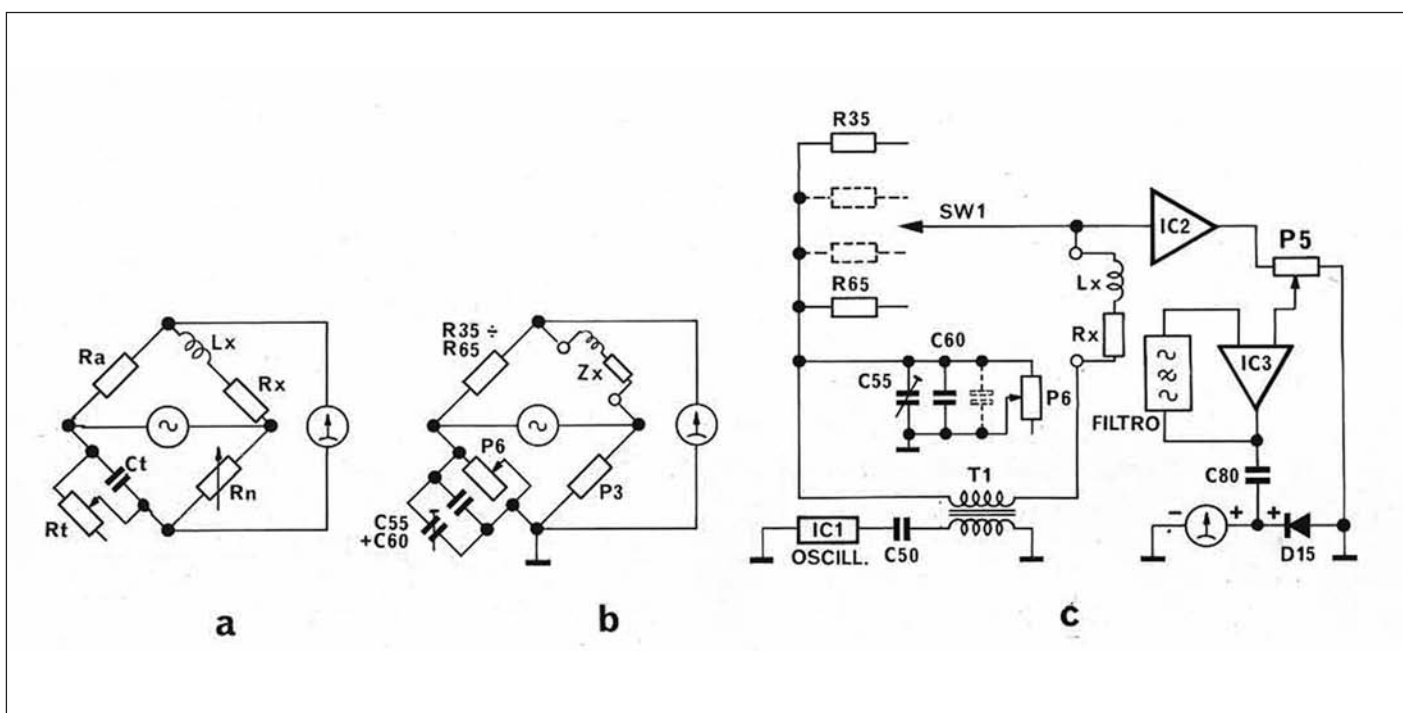



Figura a - b - c: possibili variazioni del ponte di Maxwell.

iLCD
Next Generation
Intelligent LCDs

dal microcontrollore. Veniamo ora alla descrizione dell'oscillatore raffigurato nella figura 2, a prima vista sembrerebbe un oscillatore RC che nulla centra con tutto il discorso, in realtà le cose non sono come sembrano, infatti il cuore del circuito non è la rete LC con risonanza parallelo, ma il comparatore di tensione utilizzato, un comunissimo LM311 che per la sua tecnologia è praticamente insostituibile almeno in questa applicazione. Per meglio comprendere il suo funzionamento immaginate che la rete LC non ci sia, In questa ipotesi troviamo che sull'ingresso non invertente di IC5 è presente una tensione fissa pari a Vcc/2 realizzata dal partitore di tensione costituito da R10 e R11. R8 altro non è che una retroazione positiva che provoca un isteresi nel comparatore in modo da garantire il giusto punto di lavoro onde evitare per cosi dire, esitazioni che provocherebbero delle oscillazioni non desiderate rendendo instabile il tutto.
ul piedino invertente troviamo R12 e C10 classica rete RC con retroazione positiva di R12.
Supponiamo che all'accensione l'uscita di IC1 sia positiva, cosicché grazie a R12 il condensatore C10 inizia a caricarsi fino a superare di quanto basta la soglia fissata da R10 e R11 di vcc/2, a questo punto l'uscita di IC1 commuta in negativo e il condensatore C10 comincia a scaricarsi fino a portarsi ad un livello inferiore di Vcc/2, e il ciclo si ripete. Come è facile intuire in uscita di IC1 ritroviamo un segnale ad onda quadra la cui frequenza è stabilita dalla rete R12 e C10. Ora inseriamo la

rete LC ma disaccoppiando la tensione continua dall'ingresso invertente di IC1 con un condensatore elettrolitico da 10uF vedi C9. Supponiamo che all'istante iniziale la frequenza di IC1 sia dieci volte inferiore a quella di risonanza dell'accoppiamento L1-C12, al primo impulso di IC1 una parte di questo viene riportato grazie a R8 e C9 nella rete L1-C12 che inizia ad oscillare, senza ulteriori stimoli l'onda sinusoidale generata tenderebbe a smorzarsi, ma in realtà questa piccola oscillazione viene reintrodotta sempre con C9 al comparatore che inizia ad oscillare alla frequenza di risonanza del gruppo L1-C12 e non più a quella di R12 e C10.
Questo meccanismo unito alla formidabile struttura del LM311 garantiscono un ampia escursione di frequenza e di conseguenza dei valori di Lx misurabili.
Di fatto ciò che viene misurato non è l'induttanza ma l'impedenza, poiché sappiamo bene che un induttore ha una sua resistenza finita dovuta al filo di cui è composto.
L'induttanza L1 da 100uH è di precisione come anche il condensatore C12 da 3,3nF.
La frequenza di risonanza del gruppo L1-C12 è data dalla formula che segue:

```
Freq_Osc=159000000/(L1*C12)^2
```

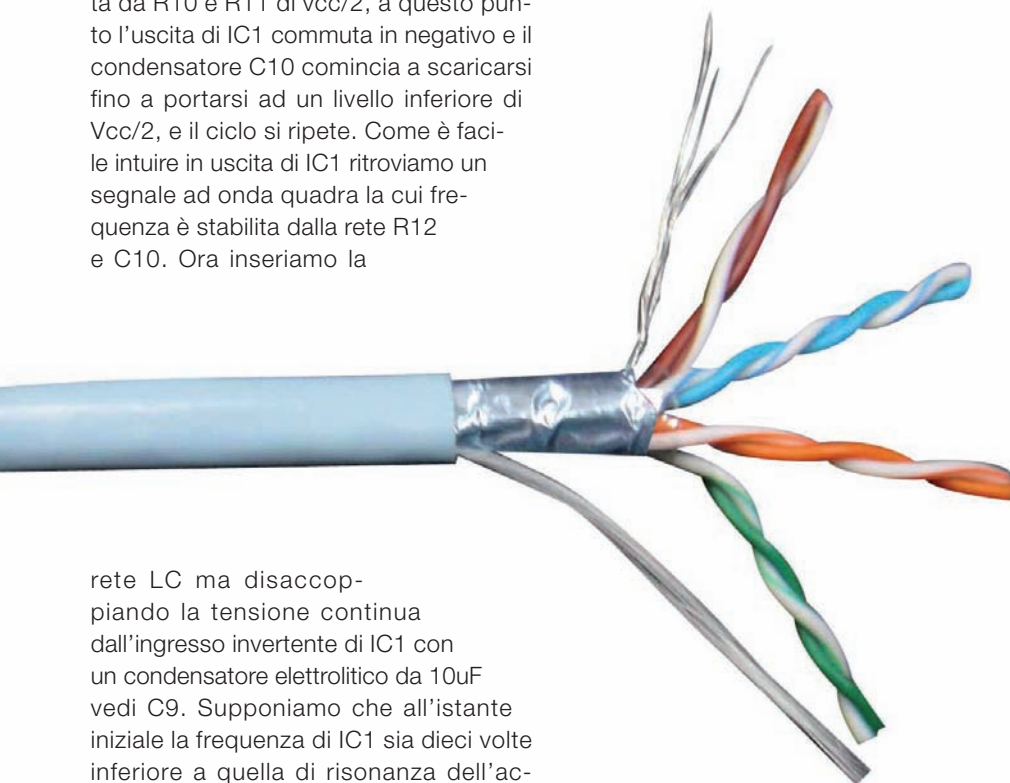
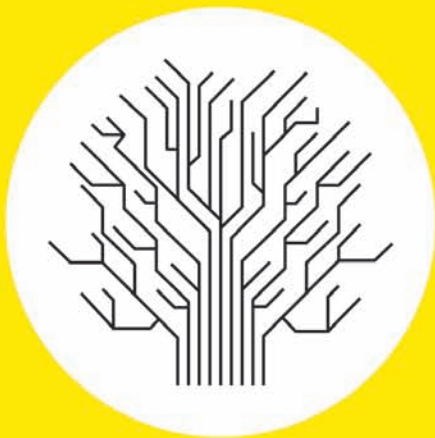
il procedimento con il quale viene effettuata la misura è semplice:

All'accensione il micro controlla T1 che attacca il relè cortocircuitando a massa L1, che cosi risulta in parallelo a C12. Allo stesso istante l'uscita out di IC1 si porta bassa accendendo il led rosso che indica l'inizio della calibrazione, a questo punto IC1 attende che venga premuto il pulsante P1 collegato in JP1 per iniziare la calibrazione. Come P1 viene premuto IC1 legge la frequenza di risonanza di L1 e C12 fissata con i valori dello schema a:

```
Freq_Osc=159000000/(100*3300)^2=276783,47 Hz
```

quindi per poter estrapolare il valore dell'induttanza in esame, e cioè 234uH il processore esegue semplici calcoli applicando la seguente formula:

```
Lx=(Freq_Osc^2)/(Freq_new^2)-1)*100=234 uH
```

Come si vede non fa altro che sottrarre 1 al rapporto dei quadrati delle due frequenza e moltiplica per 100, il risultato è esattamente il valore dell'induttanza misurata.
Quindi l'unico nostro parametro diventa la frequenza con la quale è possibile risalire al valore dell'impedenza incognita. Tenete conto che in 234uH ci sono i contributi di Rs (resistenza filo) ed eventuali capacità parassite e cosi via.
In figura 1 è presente un connettore jumper a tre poli con il quale possiamo scalare la frequenza base da 2048 a 4096, occorre cortocircuitare il 2 con il 3, in pratica il sistema scala cosi a 2048. In questa versione non è gestito il 4096, quindi non sbagliare poiché la misura sarebbe errata. Collegare a JP1 un pulsante che servirà a dare l'OK al processore per iniziare la calibrazione automatica. Utilizzare per L1 e C12 in figura 2 componenti di precisione e per L1 un fattore di merito molto alto, cosi la misura sarà ancora più attendibile. Per chi avesse domande o curiosità può contattarmi attraverso la rivista, sarò lieto di rispondere alle vostre richieste. ❑

CODICE MIP 2835006

Fiera dell'Elettronica e del Radioamatore

Fiera Nazionale.
Hardware, software,
componentistica, telefonia
ed elettronica di consumo

24 e 25 marzo 2012

dalle ore 8.30 alle 18
Padiglioni Fiera Millenaria
Gonzaga (Mantova)

CODICE MIP 2834417

fachiro.com

Evento collaterale

Fotografia 1000NARIA

Possibilità d'acquisto
e sale posa con modelle professioniste

Fiera Millenaria di Gonzaga srl

Via Fiera Millenaria 13, Gonzaga (MN)
T +39 0376 58098 F +39 0376 528153
www.fieramillenaria.it

Seguiteci su

di GIOVANNI DI MARIA

LA PORTA GPIO AL MICROSCOPIO

L'articolo descrive in dettaglio la porta GPIO del Raspberry Pi e anticipa in parte molti concetti sul suo utilizzo, che saranno esaminati nei prossimi numeri della rivista.

Il Raspberry Pi, tra le tante porte di comunicazione disponibili, ne ha una particolarmente importante, che interessa più da vicino la comunità dei progettisti elettronici, come la nostra. Si tratta della porta GPIO.
Essa può essere paragonata alle porte di I/O di un microcontrollore anche se, molte caratteristiche sono differenti. Esamineremo le funzionalità e gli utilizzi pratici di tale porta, in modo da affrontare le prossime puntate pratiche con maggiore cognizione di causa e maggiore sicurezza e disinvoltura.

Figura 1: La porta GPIO del Raspberry Pi

LA PORTA GPIO

GPIO è un acronimo e significa General Purpose I/O (Input e Output per l'utilizzo generale). Si tratta di un connettore maschio, montato su piastra, contrassegnato sul Raspberry Pi con P1, formato da 2 righe di 13 pin ciascuna, per un totale di 26 contatti.
Guardando il Raspberry Pi dall'alto, in modo che i connettori USB siano posizionati sulla nostra destra, possiamo scorgere il connettore GPIO sul lato superiore, a sinistra.
La numerazione dei pin avviene contando alternativamente le due file dei contatti, partendo dal primo pin in basso a sinistra, procedendo verso l'alto e poi proseguendo in basso a destra, sino a considerarli e conteggiarli tutti. I numeri pari sono situati in alto mentre i numeri dispari si trovano in basso. Seguite sempre un datasheet o un pinout in fase di progettazione, onde evitare banali errori nelle connessioni. E' interessante notare che tale porta, oltre a mettere a disposizione dei pin per l'ingresso e l'uscita, offre anche l'accesso ai protocolli I²C, SPI e UART.

Figura 2: Dettaglio ingrandito della porta GPIO

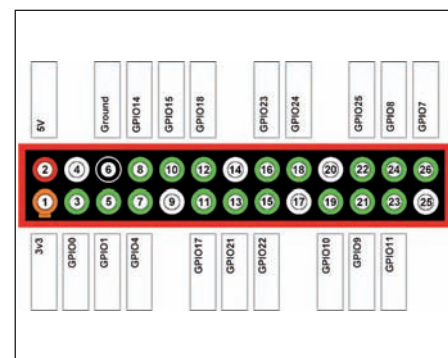



Figura 3: Numerazione della porta GPIO

EFFETTUARE UNA CONNESSIONE FISICA ALLA GPIO

Eliminate a priori l'idea di effettuare il collegamento alla porta GPIO tramite fili volanti e prese con pinzette a coccodrillo. I contatti hanno infatti un passo standard di 2,54 millimetri (1/10 di pollice) ed un corto circuito o un contatto accidentale con altri fonti di corrente potrebbe distruggere l'intero sistema. Il passo del connettore non è particolarmente ridotto, ma è meglio non sentirsi troppo sicuri. Non collegare mai qualcosa di pericoloso al Raspberry Pi.

La maggior parte dei pin della GPIO è collegata direttamente al chip Broadcom. Occorre quindi prestare la maggiore cura possibile e la massima attenzione, in fase di progettazione dei circuiti esterni da collegare al Raspberry Pi, in quanto è molto facile distruggerlo, per colpa di una svista o di un piccolo errore. Il nostro motto è quello di controllare tante volte il circuito, anche quando si è sicuri che tutto è corretto. Si può utilizzare un tester per controllare la presenza di eventuali corto circuiti accidentali.

Quando si lavora con la porta GPIO del Raspberry Pi, si deve sapere esattamente cosa si sta facendo e si deve anche conoscere la funzione svolta da ciascun pin. Se durante le prove e gli esperimenti con la GPIO si nota qualcosa di "strano" nei comportamenti del Raspberry Pi è consigliabile spegnere immediatamente il sistema, onde prevenire danni irreversibili. Per la connessione il sistema migliore è quello di utilizzare un cavo piatto composto da 26 conduttori, simile a quello che si usa per il collegamento di un Hard Disk. Se si prevede di utilizzare solamente un numero limitato di porte della GPIO, si possono utilizzare i cavetti colorati unipolari e bipolari, che si usano per connettere la piastra madre del personal computers ai vari Led di segnalazione, allo speaker, all'interruttore di accensione e così via.

Figura 5: Cavetti di una piastra madre utilizzabili con il Raspberry Pi

Un'altra soluzione, se non si dovesse reperire il cavo piatto da 26 poli, è quella di utilizzare un normale cavo di collegamento per Hard Disk o di un Floppy Disk ed eliminare i connettori superflui. Hanno "quasi" lo stesso passo, ma la differenza è davvero minima.

LA FUNZIONE DEI PIN

In figura è riportata la mappatura della porta GPIO, con i nomi e le funzioni dei pin. Essa è riferita alla posizione frontale del Raspberry Pi, guardandolo in modo che i connettori USB siano posti sulla destra di chi guarda.

I nomi dei pin più vicini al connettore (in blu scuro) rappresentano la loro funzione mentre la numerazione più esterna (in verde scuro) classifica le porte secondo un conteggio riferito all'indirizzamento dell'interfaccia */sys/class/gpio* (vedi puntate pratiche successive).

Quando si legge lo schema della GPIO, occorre prestare la massima attenzione ai nomi. Per esempio, prendendo come riferimento il pin 11, essa accede alla porta 17 come GPIO ma il suo nome di battesimo è GPIO 0. Dunque occhio ai datasheet.

Spesso si genera molta confusione per la corretta interpretazione della piedinatura della porta GPIO. Per confondere ancora di più le idee, i pin GPIO a volte sono ri-

Figura 4: Un cavo piatto a 26 poli

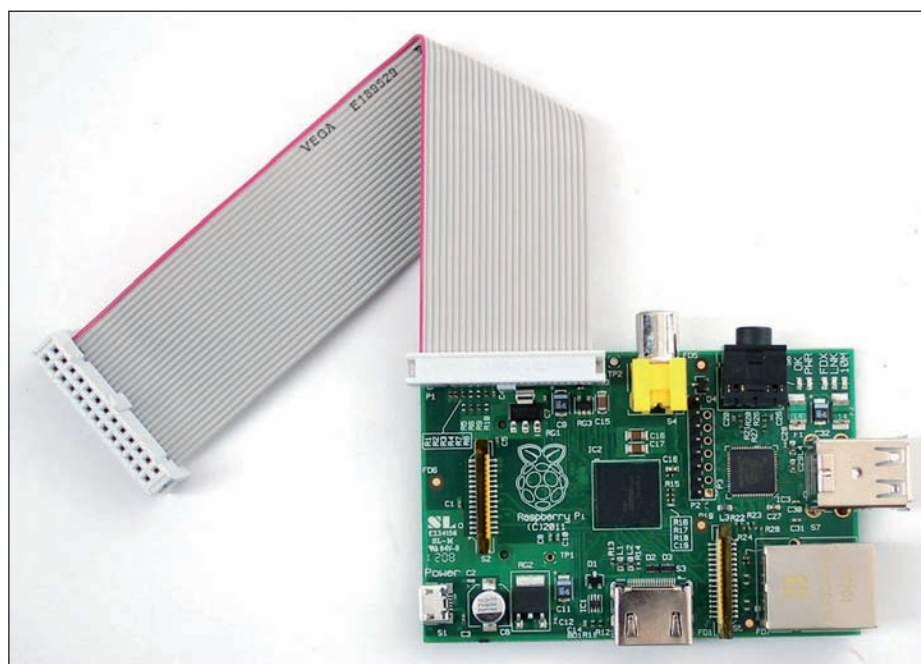



Figura 6: la porta GPIO connessa tramite cavo piatto dedicato

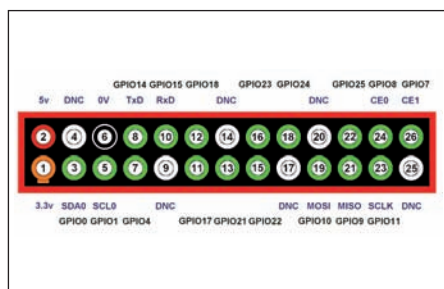



Figura 8: Nomi e funzioni dei pin della porta GPIO

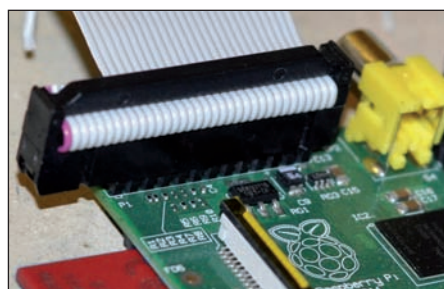



Figura 7: la porta GPIO connessa tramite cavo piatto "ridotto"

nominati con un altro insieme di numeri. Al fine di evitare di danneggiare il Raspberry Pi bisogna essere sicuri di utilizzare i pin corretti e di studiare bene la esatta connessione con altri dispositivi esterni.
Le porte contrassegnate con DNC (acronimo di Do Not Connect = non connettere) non devono essere collegate ad alcun punto del circuito. Esse sono state previste per utilizzi futuri.

I pin di tensione

Il Raspberry Pi dispone di tre pin di tensione:

- Il pin 1 fornisce una tensione positiva di 3.3V. Da essa si possono prelevare al massimo 50 mA di corrente;
- Il pin 2 fornisce una tensione positiva di 5V. E' possibile alimentare il Raspberry Pi fornendo tale tensione a questo pin ma, ovviamente, non si deve usare l'alimentatore esterno con la connessione "micro USB". Se il sistema viene alimentato tramite l'alimentatore esterno, il pin 2 eroga la tensione di 5V, con una corrente massima prelevabile di quella dell'alimentatore utilizzato, meno una quota fissa di corrente di 700mA, utilizzata dal Raspberry Pi stesso.
- Il pin 6 è la massa del sistema e deve anche essere collegato ad eventuali circuiti esterni.

I pin in oggetto sono bidirezionali, pertanto possono essere programmati ed utilizzati tanto come uscita che come ingresso.
Le uscite erogano una tensione logica di 3.3V (livello logico alto) o di 0V (livello logico basso). Se occorre una tensione TTL compatibile a 5V, è possibile utilizzare gli integrati traslatori di livello o dei semplici transistor.
Sono otto le porte considerate come vere e proprie GPIO. Le altre nove possono anche espletare funzioni molto più complesse, oltre naturalmente quelle di ingresso e uscita.
Si hanno a disposizione dunque ben 17 porte GPIO. Per utilizzarle tutte quante occorre configurare coma tali anche le porte GPIO14 e GPIO15 (pin 8-10), che di default gestiscono la porta UART.
Tutti gli ingressi funzionano con una tensione di 3.3V. Non sono ammesse tensioni superiori, pena la distruzione del sistema. Esso infatti non dispone di una protezione a monte. Le protezioni devono essere implementate dal progettista. Per utilizzi professionali si consiglia di allestire una protezione esterna con buffer e optoisolatori, onde prevenire qualsiasi danno accidentale.
Per default, tutte le 17 porte sono configurate come ingressi, tranne le due porte GPIO14 e GPIO15. Se tali pin devono

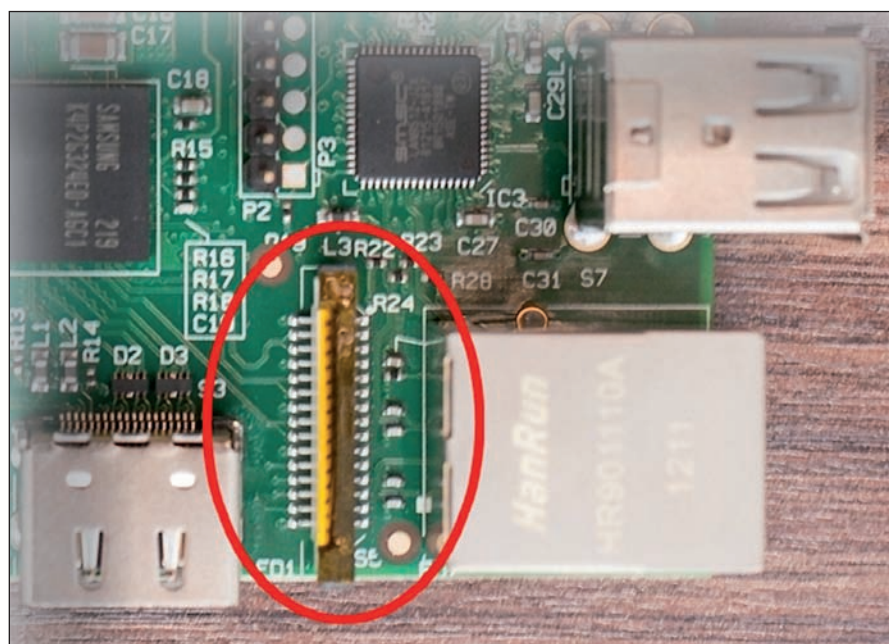



Figura 9: Dal connettore S5 è possibile utilizzare altre 4 porte GPIO

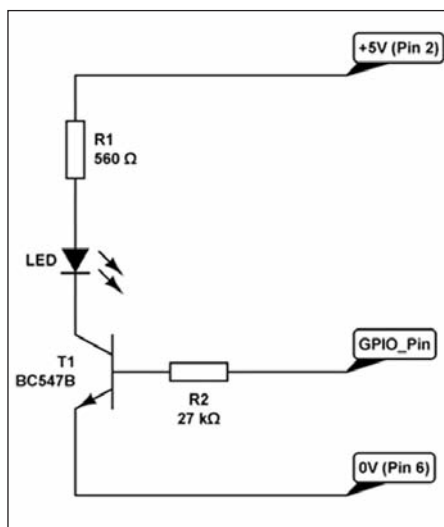



Figura 10: Un esempio di collegamento di una porta GPIO ad un transistor

Figura 12: Esempio di collegamento di poche porte della GPIO

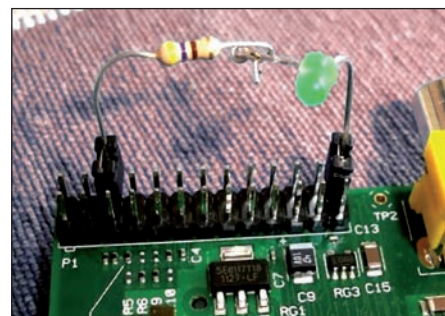



Figura 13: Collegamento volante sulla GPIO con dei jumper (non consigliato)

Figura 11: Una possibile soluzione per il collegamento alla GPIO

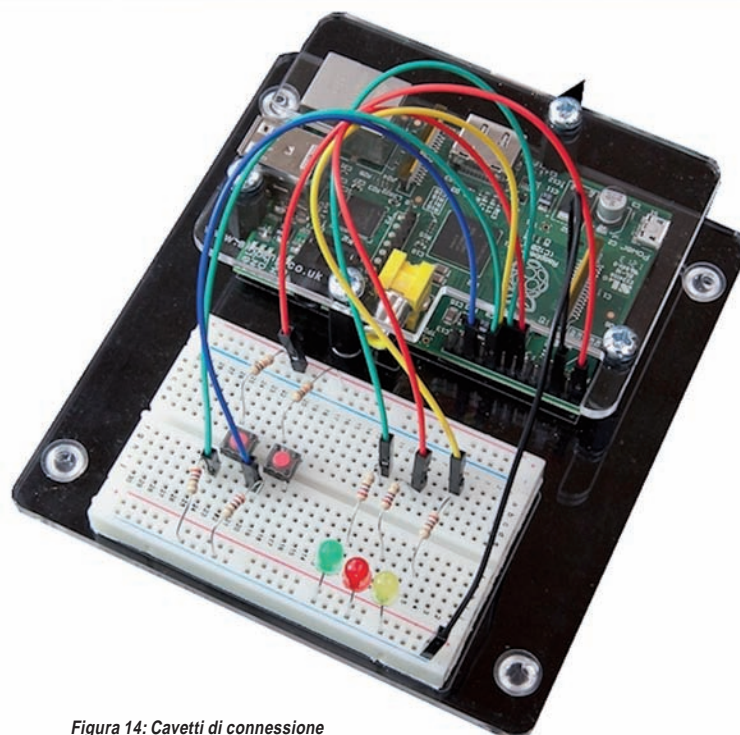



Figura 14: Cavetti di connessione ai vari pin della GPIO

essere utilizzati in un progetto, occorre configurarli correttamente in fase di programmazione.
I pin 3 e 5 del Raspberry Pi (rispettivamente le funzioni SDA0 e SCL0) possono utilizzare il protocollo I²C. Utilizzano infatti due resistenze di pull-up di 1.8Kohm. Infatti, al momento del boot, esse non si trovano a massa ma erogano una tensione di circa 1.9V. Per utilizzi pratici o professionali o produzione di centraline e prototipi, sconsigliamo di utilizzare queste due porte in uscita in quanto, al momento dell'attivazione del sistema (e durante il boot), potrebbero attivare eventuali carichi connessi. Una soluzione al problema consiste nel temporizzare i circuiti a valle oppure accenderli dopo che il Raspberry Pi ha eseguito il programma applicativo di controllo e ha disattivato le porte impiegate. Infine il pin 12 supporta anche la modulazione PWM.

17 porte sono poche?

Se le 17 porte del connettore GPIO (P1) non dovessero bastare è possibile usufruire di alcune porte aggiuntive, rintracciabili sul Raspberry Pi.
Per esempio, è possibile utilizzare altre quattro porte, come GPIO2, GPIO3, GPIO5 e GPIO27, disponibili sul connet-

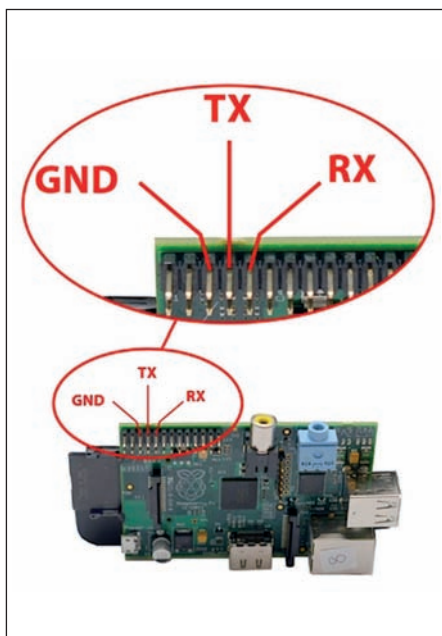



Figura 15: I terminali della UART

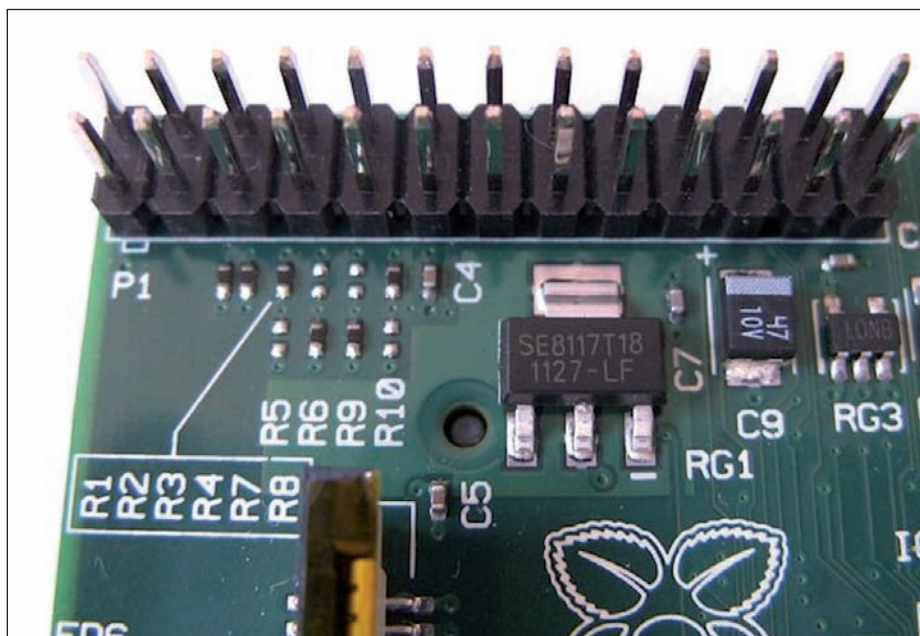



Figura 16: Il connettore GPIO

Figura 17: GPIO pinout

tore S5, formato da 15 pin. Tale connettore è fruibile quando non si utilizza alcuna camera. Altre porte sono localizzate sul PCB del Raspberry, ma sono riservate ad altre funzioni:

- GPIO16 pilota il diodo Led D5, utilizzato come indicatore di accesso della SD;
- GPIO28-31 sono utilizzate per l'ID della piastra;
- GPIO40 e GPIO45 è utilizzata per l'audio analogico;
- GPIO46 è utilizzata per la rilevazione dell'HDMI;
- Da GPIO47 a GPIO53 sono utilizzate per l'interfaccia della SD.

LA PORTA UART

Il Raspberry Pi implementa nella porta GPIO anche una connessione seriale. Essa, chiamata più precisamente "mini-uart" fa capo ai pin 8 (TX) e 10 (RX) con ovviamente la massa al pin 6 (GND). Sono imminenti alcuni articoli pratici sulle modalità di collegamento ed interfacciamento della porta seriale ad altre periferiche esterne e anche vero il PC.
Quando si collega una porta seriale sulla GPIO del Raspberry Pi, occorre considerare una complicazione: le tensioni di lavoro presenti sulla GPIO non sono le stesse di quelle richieste dagli standard RS-232 e RS-485. Infatti, come abbiamo visto prima, i pin del Raspberry Pi forniscono un livello logico di 0V e di 3.3V.
La porta RS-232 "lavora" con una tensione compresa tra -5V e -15V per il livello logico 1 (alto) e tra +5V e +15V per il livello logico 0 (basso) pertanto occorre traslare i livelli logici del Raspberry Pi, utilizzando appunto idonei traslatori di livello come, ad esempio, il circuito integrato MAX-3232CPE. Esso ha la funzione di adeguare i livelli in ingresso per ottenere le tensioni opportune. Se si lavora con lo standard RS-485, sono richiesti altri tipi di circuiti integrati. Vi sveliamo una chicca: le due porte GPIO14 e GPIO15 (pin 8 e 10), per default, sono configurate come linee seriali. Esse possono essere collegate ad un terminale seriale, per potere accedere al sistema, senza l'ausilio del TV.

CONCLUSIONI

Dalle prossime puntate inizieranno finalmente le attività pratiche, con la pubblicazione di preziosi tutorials, che insegneranno a programmare le porte in uscita e in ingresso. Non perdete pertanto i prossimi numeri, poiché il bello deve ancora arrivare. Alle prossime. ❑

CODICE MIP 2835002

Protocol Simulator Board

Problemi per analizzare i segnali BUS presenti nel proprio progetto?
Zeroplus viene in aiuto con questa scheda per l'analisi dei protocolli!
Durante lo sviluppo o la decodifica dei segnali, la scheda "PSB" aiuta a visualizzare i pacchetti e ridurre così i tempi di sviluppo migliorando l'efficienza. Lavorando con il Logic Analyzer, si possono valutare le relazioni della base di tempo e dei pacchetti migliorando il progetto.
Ideale sia per i progettisti professionisti che per la didattica.

CODICE MIP 2835507

PC System	Memory	Digital Audio	IC Interface	Optoelectronics	Wireless	Other
Compact Flash 4.1	1-Wire, 3-Wire	AC97	HPI	7-SEGMENT LED	Differential Manchester	BDM, DS 1302
eMMC	I2C	HD Audio	JTAG 2.0	CCIR656	DigRF	DS18B20
FWH	I2C(EEPROM 24L)	HDMI CEC	MCU-51 DECODE	COME IMAGE	ISO7816 UART	DSA Interface
Low Pin Count	I2C(EEPROM 24L	I2S	MICROWIIRE	DALI Interface	KEELOQ Code Hopping	GPIB, HART
LPC-SERIRQ	CS6/24LCS62)	PCM	SIGNIA 6210	DM114/DM115	MANCHESTER	ModBus
LPT, PCI, PECI	MICROWIRE(EEP	S/PDIF	SLE4442	DMX512	MII	MODIFIED SPI
PS/2	ROM 93C)	STBus	SSI Interface	LCD12864	MIL-STD-1553	OPENTHERM 2.2
DS2.0/DSIO, SVID	SPI, SPI PLUS		ST669	LCD1602	MILLER	PSB Interface
UART(RS232C	SPI(EEPROM AT25F)			LG4572	MODIFIED MILLER	PROFIBUS
/422/485)	SAMSUNG			S2Cwire/AS2Cwire	SWP	SHT11, YK-5
USB 1.1	K9(NAND flash)			SCCB	WIEGAND	1-Wire(Advanced)
USB 2.0	UNI/O				WWV/WWVH/WWVB	

Infrared rays	Basic Logic Application	Power	Automotive
IRDS	ARITHMETICAL LOGIC	BMS, HDQ	CAN 2.0B
NEC PD6122	DIGITAL LOGIC	PMBus 1.1	DSI Bus
Philips RC-5	JK FLIP-FLOP	SDQ	FlexRay 2.1A
Philips RC-6	UP DOWN COUNTER	SMBus 2.0	LIN 2.1
PT2262/PT2272			MVB
			WTB

LAP-C Series

LAP-B Series

E' un prodotto distribuito da

Per maggiori informazioni **www.elettroshop.com/psb** oppure chiamare lo **02/66504794**

di MARCO CARMINATI

SENSORI: *elaborazione* DEI SEGNALI

(parte seconda)

Completiamo con questa seconda parte la panoramica sulle tecniche di elaborazione dei segnali acquisiti dai sensori, analizzando in particolare i filtri digitali non lineari come il filtro mediano. Introdurremo inoltre i rudimenti di un'importante famiglia di strumenti matematici rappresentata dagli "osservatori", presentando in particolare il filtro di Kalman che permette di implementare digitalmente dei sensori virtuali e fondere in modo ottimo dati eterogenei, correlati con la medesima grandezza incognita.

FILTRI NON LINEARI

Tutti i tipi di filtri discussi nella prima parte dell'articolo (numero 330), sia analogici che digitali, erano di tipo lineare (e *tempo-invariante* ovvero i campioni acquisiti vengono moltiplicati per dei coefficienti costanti e sommati tra loro). I filtri lineari sono i più diffusi e generalmente preferibili per la ben consolidata teoria di dimensionamento. Esiste però la possibilità di eseguire delle operazioni tempo-varianti (nel caso in cui i coefficienti variano nel tempo) e soprattutto *non lineari* sui campioni acquisiti, al fine di ottenere particolari prestazioni dinamiche. L'esempio più comune di operazione tempo variante è la traslazione in frequenza che si ottiene moltiplicando il segnale di ingresso per un segnale sinusoidale, al fine di traslare lo spettro di ingresso attorno alla frequenza di tale sinusoide, ove la catena di amplificazione può operare in condizioni migliori (ad esempio minore rumore come nelle architetture di tipo *lock-in*).

Figura 1: Esempio di applicazione del filtro mediano ad un'immagine.

FILTRO MEDIANO

Un filtro non lineare molto interessante per sua semplicità ed efficacia è il filtro mediano. Sviluppato principalmente nell'ambito dell'elaborazione digitale delle immagini (che costituiscono segnali digitali bidimensionali) può essere applicato, con le opportune attenzioni, anche nel più semplice contesto sensoristico. Nel caso di un segnale monodimensionale (una sequenza di valori campionati) il filtro mediano opera su una finestra mobile di campioni (analogamente al filtro a media mobile) calcolando il valore del nuovo dato filtrato come la *mediana* dei dati nella finestra invece che la media. La mediana si ottiene semplicemente ordinando in modo crescente i valori presenti nella finestra ed estraendo il valore al centro della sequenza. Una possibile implementazione di una funzione che esegue la mediana di un vettore di numeri, scritta in C, è riportata nel riquadro 1. Appare così evidente che il filtro mediano esegue anch'esso un filtraggio passa basso, andando ad eliminare i valori estremi. Data la natura non lineare del filtro (il riordino dei dati) non è però possibile darne una rappresentazione nel dominio di Fourier né calcolarne la banda passante. Questo filtro viene tipicamente impiegato per ripulire il segnale dal rumore impulsivo, chiamato rumore "sale e pepe" nel contesto dell'elaborazione delle immagini, in quanto si manifesta in un insieme di singoli pixel distribuiti casualmente e saturi al minimo (nero) o al massimo (bianco) dell'intensita luminosa. La **figura 1** mostra un esempio di applicazione del filtro mediano ad un'immagine corrotta da un intenso rumore impulsivo (40%). Nel primo caso viene impiegata una finestra bidimensionale di 3x3 pixel, mentre nel secondo, maggiormente efficace, la finestra è 5x5. In questo caso bidimensionale, il valore del pixel centrale diventa la mediana dei valori dei pixel circostanti all'interno della finestra quadrata. Nonostante un'inevitabile perdita di dettagli rispetto all'immagine originale, la ricostruzione è ottima considerando che in presenza di rumore l'immagine è quasi irriconoscibile e, soprattutto, a valle del filtraggio i contorni rimangono sufficientemente netti. Maggiore è l'ampiezza della finestra, maggiore è il tempo impiegato nell'ordinare i valori. Il vantaggio principale offerto dal filtro mediano, rispetto al passa basso a media mobile tradizionale, è che preserva maggiormente il contrasto dei contorni nelle immagini, evitando lo smussamento dei bordi. Questa proprietà può essere sfruttata anche nell'elaborazione dei segnali di tensione monodimensionali. Un esempio dell'efficacia di questo approccio è illustrato in **figura 2** in cui vegono confrontati in una simulazione numerica un filtro mediano ed un filtro a media mobile con la stessa larghezza di finestra applicati ad un segnale a scalino (di ampiezza unitaria) affetto da un rumore gaussiano con varianza pari al 0.1V. Come si può notare, il segnale filtrato con il filtro mediano risulta "ripulito" della stessa entità del filtro a media mobile, ma la risposta allo scalino è molto netta: in soli 3 campioni il segnale raggiunge rapidamente il valore di regime, mentre il filtro a media mobile (FIR) smussa lo scalino in una lenta rampa di durata pari alla larghezza della finestra, ovvero 10 campioni.

SENSORI VIRTUALI

Nella maggioranza dei casi, quando si è interessati a misurare una certa grandezza fisica, si seleziona un sensore adatto a misurare direttamente tale grandezza. Se possibile, si sceglie un sensore le cui prestazioni siano adeguate alla misura (fondoscala, risoluzione, ecc...). In particolare, se la banda passante del sensore è sufficientemente maggiore della banda del segnale, cioè se il tempo di risposta del sensore è trascurabile rispetto alla scala dei tempi di evoluzione del sistema fisico, la dinamica del sensore può essere trascurata ed il comportamento del sensore è descritto da una semplice relazione algebrica. Normalmente questa relazione algebrica è lineare, almeno in un dato intervallo di valori, e quindi l'uscita del sensore è una tensione y esprimibile come $y = a + b \cdot x$, dove a costituisce un offset (spesso prossimo a zero o facilmente calibrabile), b è il guadagno del sensore e x è il misurando. Le unità di misura di k e di a sono evidentemente Volt, mentre quelle di b sono Volt per unità del misurando (ad esempio V/°C per la temperatura o V/bar per la pressione). Conseguentemente, per risalire al valore del misurando x a partire dalla tensione y, la relazione algebrica va semplicemente invertita, ottenendo $x = (y-a)/b$. Nel caso poi in cui la relazione algebrica $y = f(x)$ sia non lineare, è possibile

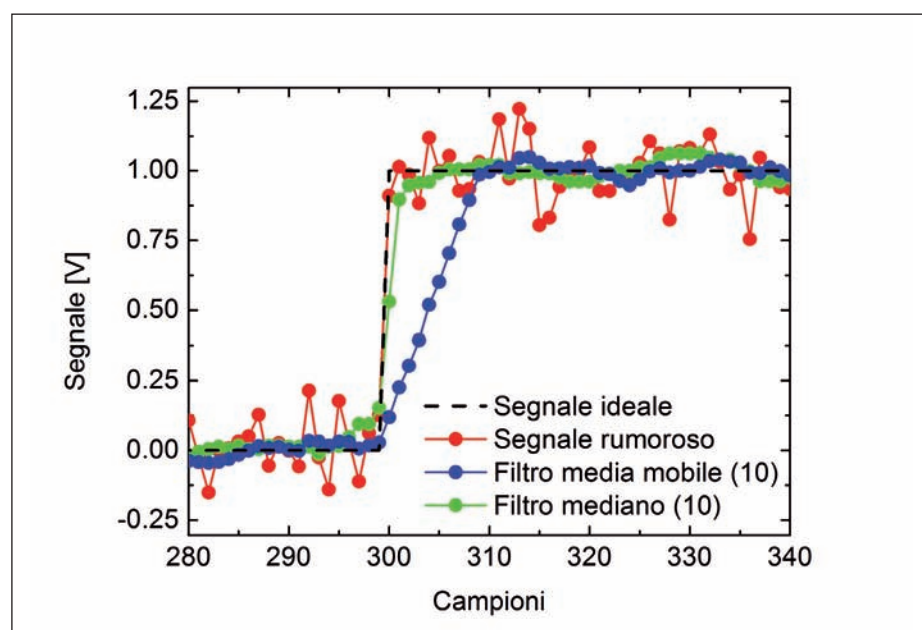



Figura 2: Simulazione digitale (in ambiente Matlab®) di confronto tra un filtro passa basso FIR ed un filtro mediano di pari finestra applicati ad un segnale a scalino.

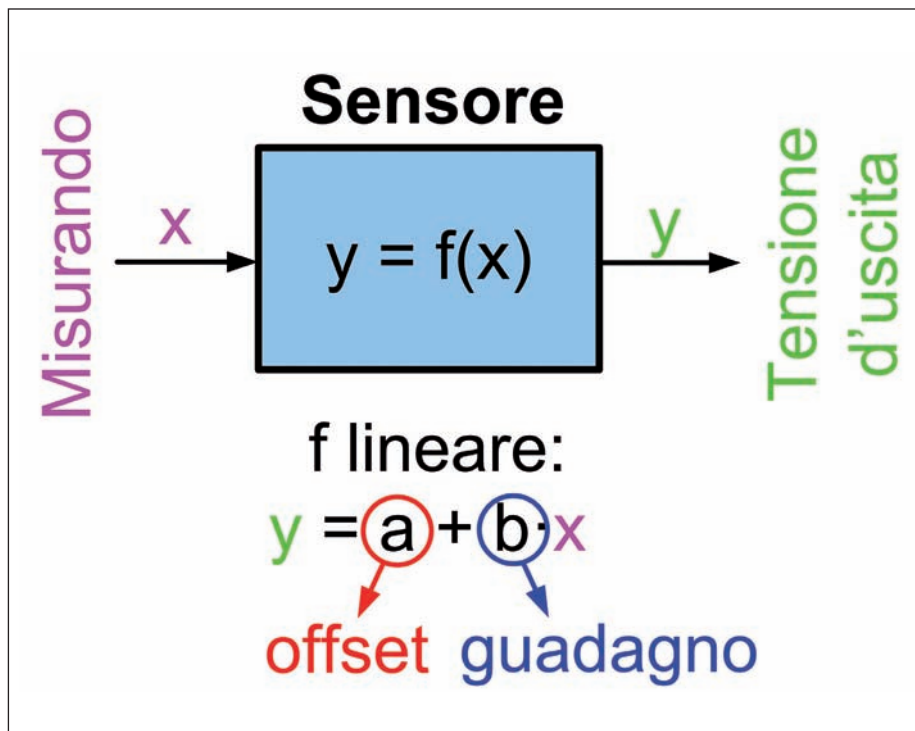



Figura 3: Funzione di trasferimento algebrica (lineare) di un generico sensore.

correggere numericamente le non linearità, ad esempio tramite una curva di calibrazione sperimentalmente ricavata (immagazzinando cioè in una memoria non volatile un set di fattori correttivi, interpolabili tra loro con un spezzata di segmenti rettilinei, per un corrispondente set di valori di y).
Un'evoluzione meno nota ma molto interessante dell'operazione di inversione di una relazione algebrica è il passaggio ad una relazione dinamica. Naturalmente il caso in cui la dinamica del sensore diventi dominante rispetto alla dinamica del misurando è certamente subottimo, in quanto la risposta del sensore si mescola con l'evoluzione dinamica del misurando (rendendo così difficile la deconvoluzione delle due) fino al punto estremo in cui il sensore può mascherare completamente la dinamica del segnale (condizione pessima ed accettabile solo per segnali quasistazionari).

L'ulteriore passaggio concettuale di estensione significativa di questo approccio si applica al caso in cui non sia possibile accedere direttamente con un sensore alla misura di un certa grandezza fisica x. Se è pero possibile misurare una grandezza w che sia dinamicamente legata a x, allora, in certe condizioni, è possibile risalire a x dalla misura di w. Un esempio molto semplice è il caso in cui si voglia conoscere la posizione di un oggetto mobile (ad esempio un robot) all'interno di un edificio e pertanto non sia possibile usare un sensore assoluto di posizione come il GPS. Se utilizziamo un accelerometro tridimensionale MEMS montato sul robot, siamo in grado di eseguire la misura dell'accelerazione nelle tre direzioni del moto lineare che sono legate cinematicamente alla posizione. Infatti, se integriamo l'accelerazione otteniamo la velocità e se integriamo la velocità otteniamo la distanza percorsa. Così, partendo da una posizione iniziale nota, è possibile ricavare istante per istante la posizione a partire dalla sola misura dell'accelerazione e dalla sua doppia integrazione. Oltre alla scomodità di dover conoscere in maniera indipendente la posizione iniziale, il difetto principale di questo tipo di misura (chiamata in navigazione *dead reckoning*), è che l'operazione di integrazione nel tempo (ovvero di addizione) produce un accumulo di errori via via divergente. Se, ad esempio, il sensore di accelerazione è affetto da un offset, questo valore costante verrà integrato dando luogo ad un errore sulla velocità che cresce linearmente con il tempo (tanto più velocemente quanto maggiore è l'entità dell'offset). A sua volta la rampa di errore sulla velocità verrà integrata dando luogo ad un errore sulla distanza percorsa che cresce nel tempo come una parabola, quindi ancora più velocemente, riducendo corrispondentemente l'intervallo di tempo massimo di utilizzo del sistema.
Questo semplice esempio ci permette di cogliere contemporaneamente i pregi ed i difetti di questo approccio: la conoscenza del legame dinamico tra due grandezze consente di realizzare dei sensori "virtuali", misurandone una (o più) per ricavare indirettamente l'altra, al prezzo di dover accettare dei compromessi nelle prestazioni.

FILTRO DI KALMAN

In molte applicazioni, posta l'esistenza del vincolo di osservabilità tra la variabile misurata e la variabile di stato che si desidera monitorare, l'implementazione di sensori virtuali è complicata da due tipi di

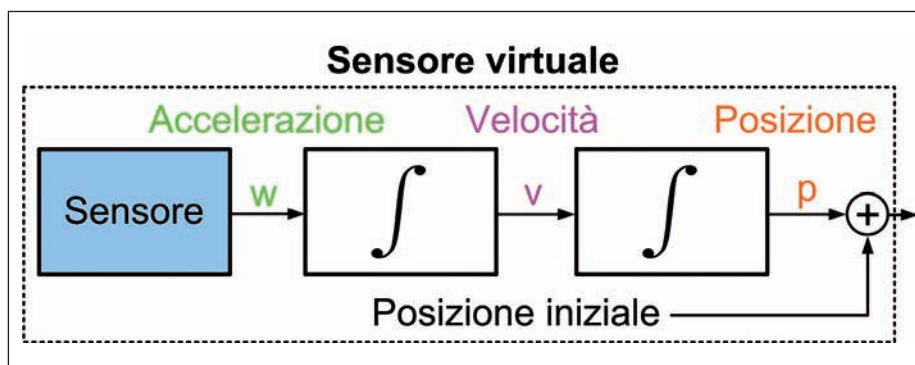



Figura 4: Esempio di un sensore di posizione virtuale ottenuto integrando due volte il segnale di un accelerometro lineare MEMS.

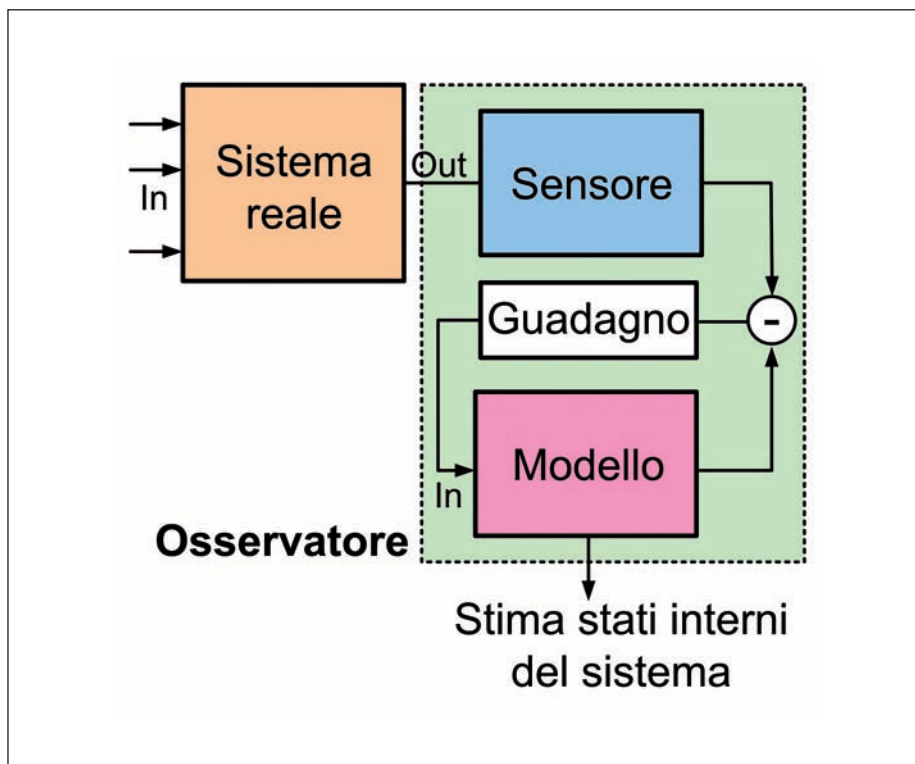



Figura 5: Architettura di un osservatore.

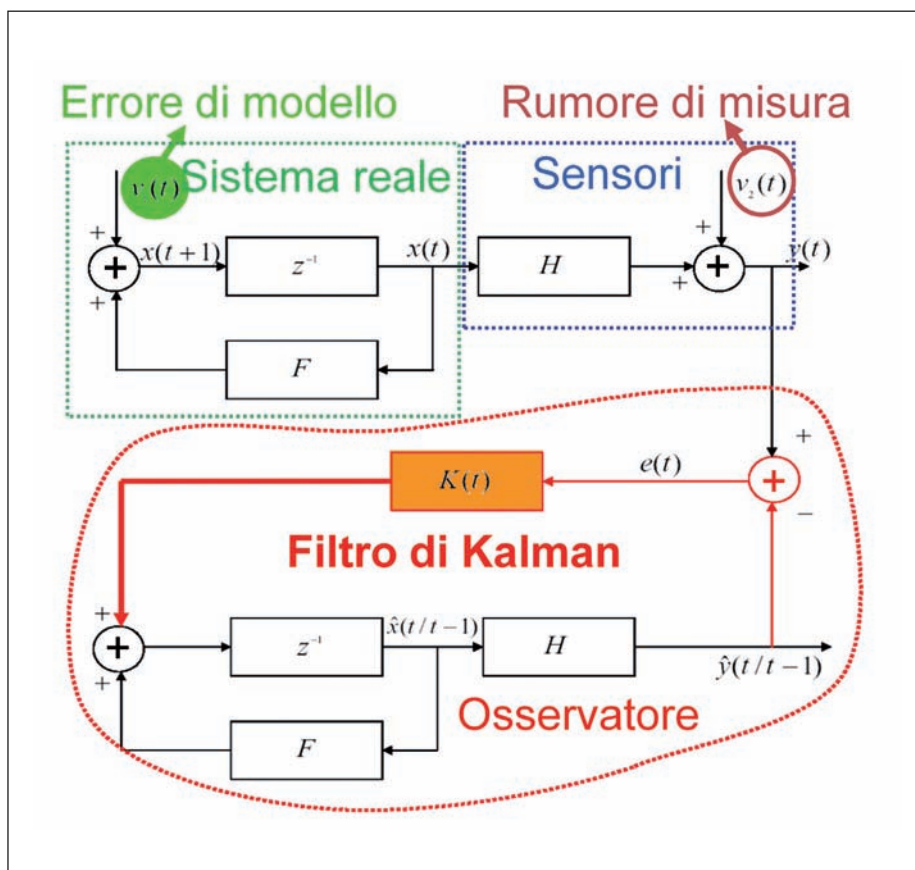



Figura 6: Schema del filtro di Kalman per l'implementazione di sensori virtuali.

non idealità. Una è l'inevitabile presenza del rumore (dovuto al sensore e/o ai disturbi ambientali) che si sovrappone al segnale, redendo a volte più difficile l'estrazione del segnale tramite l'inversione. La seconda è che non sempre sono note con precisione le equazioni differenziali che legano i vari parametri che descrivono l'evoluzione dinamica del sistema. In questi casi si costruisce un modello matematico che approssimi al meglio il comportamento del sistema fisico, cercando un compromesso tra qualità del modello (ovvero aderenza alla realtà) e complessità computazionale (cioè numero di variabili ed equazioni). Esistono sostanzialmente due approcci opposti che guidano la creazione di un modello matematico: uno, chiamato "dei principi primi", si basa sulla scrittura delle equazioni fisiche che regolano il sistema a partire dalla conoscenza dei singoli elementi che lo compongono. Quando i dettagli sulla struttura interna del sistema non sono noti o accessibili, si procede con tecniche di identificazione "black-box" che permettono di identificare un modello a partire dall'analisi delle sole uscite misurabili. Poiché le condizioni iniziali spesso non sono note e la dinamica del sistema può essere influenzata da numerosi ingressi, l'architettura che viene adottata in questi casi è illustrata in **figura 5** ed è chiamata "osservatore". Il modello rappresenta così una replica virtuale del sistema reale: entrambi evolvono in parallelo e la differenza tra le loro uscite, che nel caso ideale dovrebbe essere nulla, viene moltiplicata per un fattore di guadagno e usata come segnale errore applicato in ingresso al modello, secondo una retroazione che quindi corregge gli scostamenti.

All'interno della famiglia degli osservatori, il filtro di Kalman rappresenta lo strumento matematico ottimo per affrontare questo tipo di problemi. Inventato dall'ingegnere e matematico Rudolf Emil Kalman negli anni 50, è un algoritmo digitale ricorsivo che consente di seguire l'evoluzione nel tempo delle variabili di stato interne di un sistema dinamico a partire dalla misura di alcune uscite misurabili. Basato sempre sulla struttura osservatore/predittore contiene una replica del sistema fisico che include il modello (ovvero

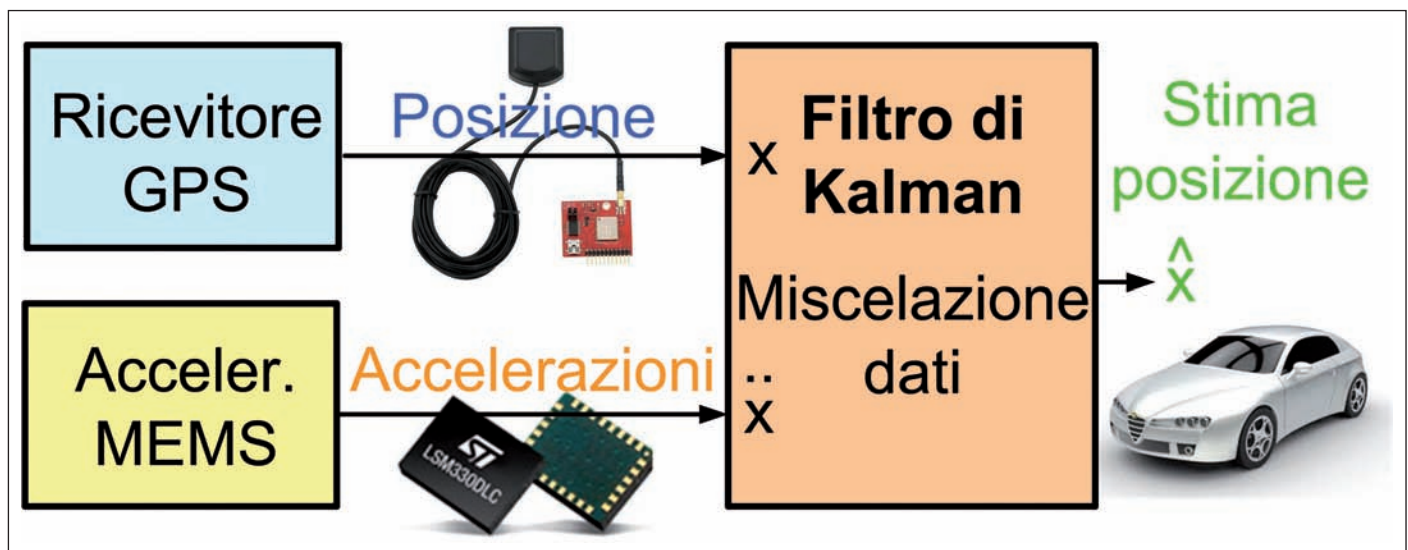



Figura 7: Esempio di architettura complementare di "fusione" dei dati che combina un accelerometro MEMS con un ricevitore GPS per una stima più affidabile della posizione di un veicolo.

la matrice F che descrive la dinamica delle variabili di stato contenute nel vettore x(t)) e la matrice H che lega le uscite misurabili (vettore y) con le variabili interne di stato. La potenza di questo filtro sta nel trovare il valore ottimo di dei guadagni correttivi K(t), ricalcolati ad ogni passo di iterazione in cui la predizione delle nuove uscite stimate y viene confrontata con la misura reale. Questo valore dipende da come evolve nel tempo la "fiducia" che si ha nella qualità delle misure (affette da un rumore fisico v_2) e dalla fiducia che si ha nell'accuratezza del modello (rumore virtuale v_1 che alimenta il sistema). Nella sua forma base, il filtro di Kalman opera con sistemi lineari, ma può essere applicato anche a sistemi non lineari, linearizzando le equazioni ad ogni iterazione rispetto ai valori del passo precedente (filtro di Kalman esteso).

Infine questo algoritmo viene usato anche per "fondere" dati eterogenei allo scopo di migliorare la stima di una certa grandezza fisica miscelando le uscite di più sensori che misurano grandezze ad essa correlate. La miscelazione si ottiene costruendo un'opportuna matrice H che leghi le uscite dei molteplici sensori alla variabile di stato di interesse (e/o alle sue derivate). Ritornando all'esempio dell'accelerometro MEMS, questo potrebbe essere utilizzato in combinazione con un ricevitore GPS, come sensore ausiliario, per stimare la posizione di un veicolo nei periodi in cui sia assente segnale GPS (ad esempio nelle gallerie). Nel breve termine la doppia integrazione dell'accelerazione fornisce la posizione del veicolo con grande risoluzione temporale e spaziale, mentre il GPS fornisce periodicamente la posizione assoluta, "resettando" gli errori accumulati nell'integrazione ed operando così in maniera perfettamente complementare (**figura 7**). ❑

RIQUADRO 1 CODICE IN C CHE IMPLEMENTA LA MEDIANA

```
% Filtro Passa Basso FIR –
4 Taps
Inizializzazione
X1 = X2 = X3 = X4 =0;
loop{
  Acquisisci Xnuovo;
  X4 = X3;
  X3 = X2;
  X2 = X1;
  X1 = Xnuovo;
  Ynuovo = 0.25*(X1+X2+X3+X4);
};
% Filtro Passa Basso IIR –
4 Taps
Inizializzazione
Yold =0;
loop{
  Acquisisci Xnuovo;
  Yold = Ynuovo;
  Ynuovo = 0.25*Xnuovo +
0.75*Yold;
};
```

PER *approfondire...*

- S. Bittanti, "Identificazione dei modelli e sistemi adattativi", Pitagora Editrice, Bologna 2003.
- D. G. Luenberger, "An Introduction to Observers", IEEE Transactions on Automatic Control, vol. 16, pp. 596-602, 1971.
- R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", Transactions of ASME, vol. 82, pp. 35-45, 1960.

CODICE MIP 2834851

Interfacciamento dei processori *(parte quindicesima)*

Display a matrice CON CONTATORE JOHNSON

La scelta di coinvolgere macchine combinatorie (decoder binari) nella gestione dei Display a Matrice di punti ha consentito di ottimizzare il numero di linee necessarie in questi casi; ma ci si può far di meglio? La risposta sta nell'utilizzo delle macchine sequenziali, come il misterioso Contatore Johnson.

Nelle puntate precedenti abbiamo ottimizzato le risorse necessarie per assicurare l'interfacciamento dei display a matrice di punti affidando la gestione delle sue 5 (o 8) Colonne alle uscite di un decoder binario e le sue 7 (o 8) Righe alla PortA di un single-chip o al Registro d'uscita Dati 0378H/0278H della porta parallela del PC.
Le scelte operate hanno garantito, con sole 10 linee d'uscita, la tecnica di multiplexing dell'informazione necessaria per un display 7x5 e il controllo di ben 3 display 7x5 o di 2 display 8x8, rispettivamente con solo una o due linee in più: non c'è da stupirsi: la macchina combinatoria coinvolta (un decoder binario) è specializzata in questo compito, essendo intrinsecamente un demultiplexer; ma ci si può chiedere se si può far di meglio.
La risposta è quella di utilizzare macchine sequenziali: solo così è possibile ridurre drasticamente il numero di linee necessarie sia per la gestione delle Righe che per quella delle Colonne; anche intuitivamente possiamo pensare a scandire i singoli bit da associare a ciascuno dei 35 o 64 Dot LED di un display affidandoli al contenuto di registri a scorrimento (*Shift Registers*) anche se il problema di sincronizzarne l'azione (al fine di garantire l'indispensabile effetto legato alla persistenza ottica) sembra apparentemente complicato.
i Registri a scorrimento sono, con i contatori, le macchine sequenziali più emblematiche e il loro impiego può dare un deciso miglioramento nella gestione dei visualizzatori alfanumerici riducendo sensibilmente le linee necessarie per governarli, anche in presenza di molti display a Matrice; le soluzioni sono numerose e dovremo, per questo, spalmarle su più puntate ..
Quella che analizzeremo in questa fa riferimento al 4017 (*5-stage Johnson counters*), un integrato definito *contatore* ma che in realtà è un *Registro a scorrimento* dotato di particolari circuiti combinatori in grado di interpretare i bit (di volta in volta presenti sulle uscite dei flip-flops di cui è dotato internamente) in modo da offrire dieci linee d'uscita *decodificate*, attive alte.
Il suo pinout e il suo schema pratico sono visibili rispettivamente in **Figura 1** e in **Figura 2**, sintetizzando i segnali necessari per il suo funzionamento e quelli messi a disposizione per il suo corretto utilizzo; il controllo sequenziale è affidato al segnale di Clock CK, attivo sul fronte di salita, ottenuto a partire dall'onda quadra CP0, fatta passare attraverso una porta AND solo se il segnale CP1 (Clock Inhibit) è forzato a zero.

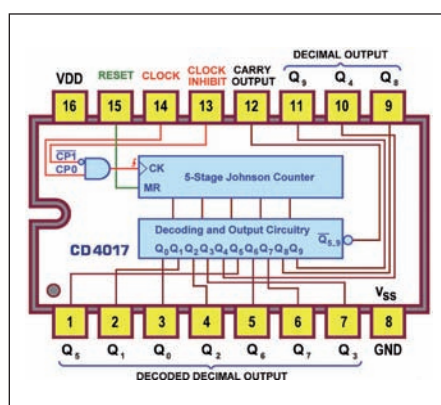



Figura 1 - 5-stage Johnson Decade Counter with 10 Decoded Outputs 4017: Pin-out

Figura 2 - 5-stage Johnson Decade Counter with 10 Decoded Outputs 4017: Schema Pratico

In realtà (sebbene il precedente punto di vista sia più razionale ..) lo stesso CK può essere ottenuto anche tenendo CP0 alto e imponendo una transizione "da alto a basso" su CP1; in ogni caso l'azione logica della porta AND, di tipo Trigger di Schmitt, restituisce impulsi formattati che consentono un buon controllo di Clock anche in presenza di fronti di salita e di discesa lenti.

Lo schema pratico è utile per evidenziare il nome dei segnali logici che entrano e di quelli che escono, nel contesto di un progetto, ma in questa occasione non aiuta a capire la misteriosa metamorfosi da *Shift Register* a *Counter* che caratterizza questo componente; in questo ci può aiutare il dettaglio funzionale proposto in **Figura 3**.

Le prime cose che saltano agli occhi sono la presenza dei cinque flip-flop D-Type di cui è dotato .. e il fatto che il *Registro a scorrimento* da essi formato è *retroazionato*, cioè l'uscita negata dell'ultimo stadio è collegata con l'ingresso seriale del primo stadio: è questo il *dettaglio* che lo trasforma in *Johnson Counter*.

Un livello alto sulla linea di Reset (MR, overriding asynchronous master reset, in comune con tutti i D-Type come la linea di Clock) azzera tutte le uscite esterne (da Q1 a Q9) meno la prima (Q0=1) indipendente dal valore corrente degli ingressi di clock (CP0, CP1): anche se gli impulsi di CK continuano ad arrivare .. lo scorrimento dei dati si è fermato.

Con MR=1 sono ovviamente azzerate anche tutte le uscite interne dei flip-flops e, non appena MR torna a 0, gli stati logici presenti su esse sono passati in sequenza da uno stadio al successivo, sul fronte di

Figura 3 - 5-stage Johnson Decade Counter with 10 Decoded Outputs 4017: Schema funzionale in dettaglio

salita di CK; in questo modo lo *Shift Register* (essendo l'ingresso D del primo stadio comunque assunto dall'uscita negata dell'ultimo) si riempie prima di 1 (spingendo fuori gli 0) e poi di 0 (spingendo fuori gli 1), all'infinito.
Queste sequenze di 5 bit vengono di volta in volta rielaborate dalla batteria di porte NOR sottostanti, per *tradurne* il valore (in una logica di decoder) su dieci linee esterne; questa è la chiave di lettura del dispositivo 4017: la sua caratteristica non è quella (tipica di un contatore) di fornire il codice binario corrispondente al conteggio progressivo di fronti attivi memorizzati durante il suo funzionamento, ma di offrire dieci linee d'uscita *decodificate*, tutte a zero meno una, quella numericamente corrispondente al valore corrente del conteggio stesso.
La **Figura 4** riassume questi concetti nel diagramma temporale delle uscite esterne (tutte con periodo di 10 cicli di CP0) e, tra queste, l'andamento del segnale Carry Output che, essendo prelevato dall'uscita interna negata del flip-flop più significativo, ha un duty cycle del 50%: esso viene posto a 1 (come l'uscita esterna Q0) dall'azione dei MR e rimane alto durante i primi cinque cicli e basso negli stati 5, 6, 7, 8, e 9: per questo è detto anche Q5_9negato.
Nonostante il suo nome è però facile capire che questo collegamento non serve e non basta per aumentare (in modo consequenziale e naturale) il numero di uscite decodificate disponibili, supponendo di utilizzarlo per pilotare l'ingresso CP0 di un eventuale 4017, posto in cascata; le uscite del secondo 4017 vanno alte (da Q"0 a Q"9) ogni volta che tutte le uscite del precedente (da Q'0 a Q'9) hanno fatto la stessa cosa: nei primi 10 impulsi di Clock è a 1 solo Q"0; nei secondi 10 impulsi è attivo solo Q"1, e così via; in queste condizioni sembra piuttosto complesso poter disporre di 100 uscite decodificate "una dopo l'altra", se non appesantendo il progetto con numerosi componenti aggiuntivi.
Il problema è accattivante ma non strettamente necessario visto che, per ora, ci basta un solo *Johnson Counter*; tuttavia, per appianare l'eventuale curiosità, è sufficiente consultare lo schema consigliato dai datasheets per aumentare il numero di uscite decodificate sequenzialmente, collegando più di un dispositivo in cascata; la **Figura 5** mostra la tecnica per estendere il numero di stati di uscita decodificati proposta da NPX Semiconductors che, nella didascalia, assicura che "le uscite decodificate sono sequenziali all'interno di ogni fase e da uno stadio all'altro, senza alcun tempo morto (tranne il ritardo di propagazione)".
Vediamo dunque, finalmente, la soluzione proposta per il controllo di un visualizzatore con una Matrice a 7x5 LED; nel progetto di **Figura 6** la gestione delle 7 Righe è affidata ancora alle linee d'uscita della PortA di un microcontrollore mentre per il controllo delle 5 Colonne sono ora sufficienti solo 2 bit della PortB: la soluzione migliore finora analizzata!
La gestione software di questo progetto è molto semplice; anzitutto è consuetudine predisporre una Tabella contenente le 5 parole da trasferire sulle Righe, una per ciascuna delle 5 Colonne previste da questo visualizzatore: ciascuna *word* (della dimensione di un byte) deve contenere i bit necessari per tenere accesi (con 1) o spenti (con 0) i rispettivi Led della Colonna ad essa relativa, e poiché le Righe previste sono sette, il valore del bit più significativo (bit7), irrilevante, potrà essere tenuto a 0.
Dopo aver predisposto il primo byte della Tabella sulla PortA è necessario fissare per qualche tempo a 1 il bit1 della PortB, per tenere brevemente a 1 anche il MR del *Johnson Counter* al fine di forzare a zero le sue uscite da Q1 a Q9 e a 1 solo la sua uscita Q0; immediatamente le Righe che risulteranno forzate a 1 (dai bit con questo valore logico presenti sulla prima porta) accenderanno i corrispondenti Led, collegati "a catodo comune" sulla prima Colonna (C1, quella a sinistra del display), messa a massa dai drives invertenti presenti nel *Darlington Transistor Array* ULN2003.
Da notare che questa situazione permane attiva per tutto il tempo nel quale MR=1 per cui, sebbene la sua durata sia irrilevante, conveniente tenerla breve e riportare velocemente a 0 il segnale di Reset MR; fatto questo, l'interfaccia 4017 è attiva e il programma deve attendere un breve istante (dell'ordine del millisecondo) e poi fornire (sul bit0 della PortB) il primo fronte attivo di Clock,

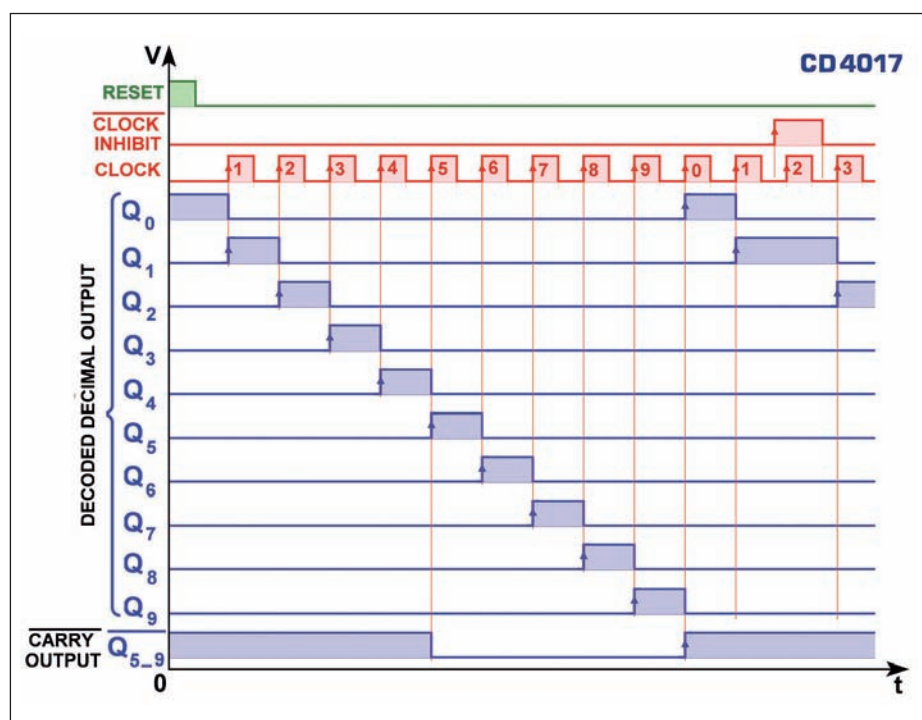



Figura 4 - 5-stage Johnson Decade Counter with 10 Decoded Outputs 4017: Diagramma temporale

subito dopo aver provveduto a scrivere sulla PortA il secondo byte previsto dalla Tabella del simbolo che si desidera visualizzare, rendendo così visibile il contenuto sulla seconda Colonna, forzata a massa dall'azione congiunta dell'uscita Q1 invertita dal driver ULN2003 ad essa collegato.
Dopo aver consumato un altro millisecondo la medesima operazione sarà condotta per ciascuno dei tre bytes rimanenti: dopo poco più di 5 ms tutte le Colonne sono state interessate dall'informazione necessaria per mostrare l'immagine "a punti" sul visualizzatore, probabilmente appena intuibile.
Solo ripetendo all'infinito tutti i comandi descritti essa sarà visibile stabilmente, per l'effetto di persistenza ottica: la commutazione continua tra le Colonne consentirà al simbolo di formarsi in modo sempre più nitido al crescere della frequenza di scansione, fino a risultare del tutto leggibile e privo di sfarfallamenti, non appena essa avrà raggiunto un valore intorno ai 50 Hz.
La creazione di impulsi è una delle cose magiche che consente la programmazione a basso livello: basta sfruttare il ciclo macchina (detto anche *Ciclo T*) che governa il processore del microcontrollore, cioè la durata del suo periodo di Clock; per esempio con un Atmel AVR ATMega168 da 32 MHz questo periodo vale 31,25 ns.
Quando una CPU esegue una qualunque delle istruzioni previste dal suo Set consuma uno o più cicli T e, senza entrare troppo nel merito, per formattare (sul bit0 di PortB) il fronte di salita necessario al nostro Clock CP0, il microprocessore dovrà eseguire una manciata di istruzioni per predisporre a 01 i due bit meno significativi di un suo registro interno, trasferirne il contenuto sulla porta, riportare a 00 il valore dei bit1,bit0 del registro e rispedirlo alla porta; tutto questo, in tempo reale, corrisponde ad un centinaio di nanosecondi.
Quando si utilizza un 4017 è opportuno mettere in evidenza una cosa importante: nel suo impiego "integrale" non è più necessario intervenire sul pin15 (MR), dopo avergli fornito *una-tantum* un breve impulso alto: non appena MR torna a 0 verranno attivate in sequenza, reiteratamente, tutte e dieci le sue linee d'uscita; sfortunatamente, nei casi come quello proposto dal nostro progetto, questo fatto terrebbe "spente" le colonne della Matrice di LED per 5 impulsi di Clock su 10, producendo una controproducente pausa di visualizzazione!
La soluzione è quella di forzare un nuovo MR prima di emettere il fronte attivo del sesto Clock e può essere operata in diversi modi; poiché l'esigenza di garantire la visualizzazione per persistenza ottica ci obbliga comunque a rifare *ad libitum* esattamente tutto il programma descritto finora, la via più logica ed economica è quella "*software*" di *rientrare* nel punto di codice in cui si fissa a 1 il bit1 della PortB (proprio per porre MR=1, riportando a 1 l'uscita Q0) ed eseguendo di nuovo tutte le istruzioni fino al termine dell'ultimo ritardo di 1ms, al servizio della quinta Colonna. In questi casi, peraltro, si persegue spesso la

Figura 5 - 5-stage Johnson Decade Counter expansion: collegamento in cascata di più 4017

via “*hardware*“ che consiste nel collegare alla linea di Reset l’uscita successiva a quella ultima necessaria, talvolta con un resistore di pull-down di 10Kohm; poiché nel nostro caso sono necessarie solo le prime cinque uscite, per ottenere un funzionamento regolare e progressivo ed per evitare che il conteggio del 4017 prosegua fino all’ultima uscita, Q9, basta collegare il pin1 (Q5) con il pin15 (MR): la tensione positiva su Q5 trasmette il suo valore al MR, obbligando a far ripartire il conteggio dalla prima uscita (Q0) e riportando immediatamente a 0 anche Q5 (e MR ad essa collegato).
La **Figura 7** mostra la variante appena descritta e ci consente qualche ulteriore considerazione: a) se si vuole continuare ad esercitare anche il controllo da software della linea di Reset MR, il rientro dall’uscita Q5 non può essere diretto e, per evitare un nefasto cortocircuito, è necessario usare una porta OR, rendendo questa alternativa antieconomica per la presenza di un nuovo integrato, 4071; b) anche la soluzione marcata con questa lettera non sembra indolore: intanto va messo in conto un possibile impercettibile disagio visivo, all’accensione, dovuto al fatto che lo stato iniziale del contatore non è prevedibile e che quindi per qualche millisecondo l’immagine sul display potrebbe essere incoerente; ma valutando serenamente la situazione essa potrebbe addirittura tornare desiderabile: in fatti, il controllo del progetto può essere esercitato addirittura con una sola (!) linea d’uscita ma per funzionare è necessaria anche una linea d’ingresso, da campionare (solo una volta, all’inizio del programma) in attesa dell’1 fornito prima o poi da Q5; in questo modo non solo viene scongiurato ogni rischio (per altro trascurabile) di visualizzazioni improprie (magari aiutato predisponendo, all’inizio sulla PortA, un byte con tutti i bit a 0), ma soprattutto viene garantita la possibilità di sincronizzare ogni fronte di Clock (attivato sul pin14 via out_DB0) con il giusto byte della Tabella del simbolo da visualizzare, cosa che altrimenti sarebbe stata impossibile.
Un’ultima considerazione: può essere utile la consapevolezza che, quando il 4017 viene azzerato, la linea d’uscita Q0 rimane a 1 fino all’arrivo di un fronte di salita di Clock sul pin14; nel contesto operativo del programma descritto, dedicato esclusivamente e continuativamente ad un Display a Matrice di 7x5 LED, la cosa è irrilevante, ma se per qualche ragione di servizio la sequenza di impulsi, dopo un Reset, venisse a mancare .. la Colonna C1 rimarrebbe permanentemente attiva mostrando fastidiosamente accesi (per altro con luminosità decisamente maggiore di quella apprezzabile durante il normale multiplexing) gli eventuali LED ad essa relativi; per questo, prima di togliere il controllo, è necessario ricordarsi di caricare la PortA con un byte nullo (.. alcuni progettisti lasciano questa linea Q0 addirittura inutilizzata).

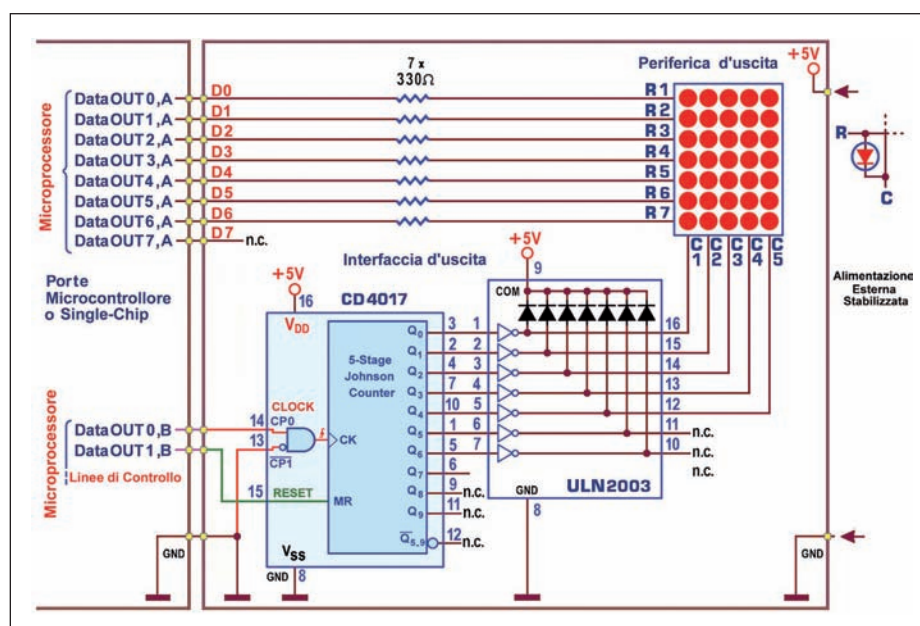
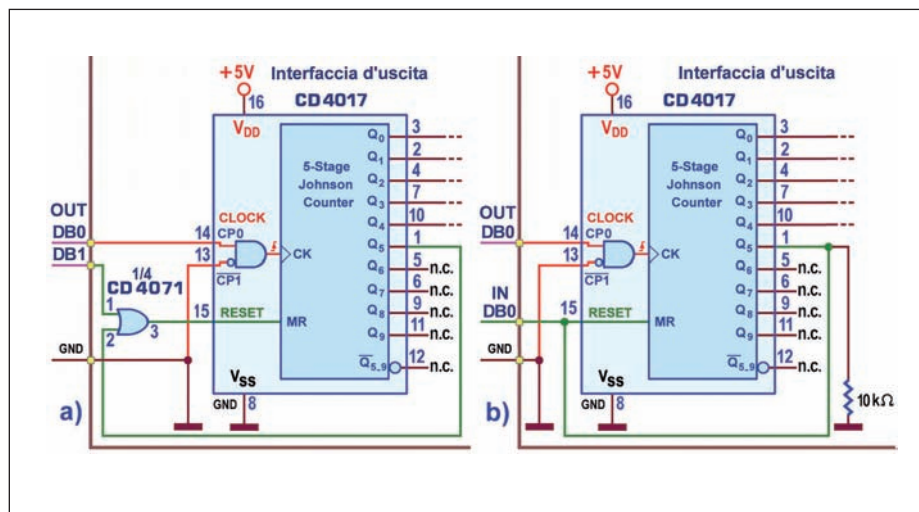



Figura 6 - Dot Matrix 7x5 LED Display: controllo con 5-stage Johnson Decade Counter 4017 tramite microcontrollore

Figura 7 - Varianti per il controllo con microcontrollore della sequenza di un 5-stage Johnson Decade Counter 4017

Il progetto appena descritto è utilizzabile anche con la porta parallela (anche SPP) di un Personal Computer: il Registro d'uscita Dati 0378H/0278H si occuperà delle 7 Righe e quello di Controllo 037AH/027AH metterà a disposizione le 2 linee per il Clock e il Reset del *Johnson Counter*; volendo optare per la soluzione che richiede la necessità di leggere un bit .. è sempre possibile far riferimento a uno dei 5 bit accessibili in ingresso tramite il Registro di Stato 0379H/0279H.

Inutile aggiungere che, con banali modifiche apportate al programma di gestione, il progetto di **Figura 7** rimane valido anche per il controllo di una Matrice a 8x8 LED. Terminiamo questa lunga trattazione fornendo un progetto in grado di sfruttare al massimo le disponibilità di un 4017, volutamente proposto con gestione via porta parallela, ma ovviamente utilizzabile anche con microcontrollore: si tratta della soluzione adatta ad un visualizzatore con due Matrici a 7x5 LED, proposta in **Figura 8**. ❏

CODICE MIP 2835686

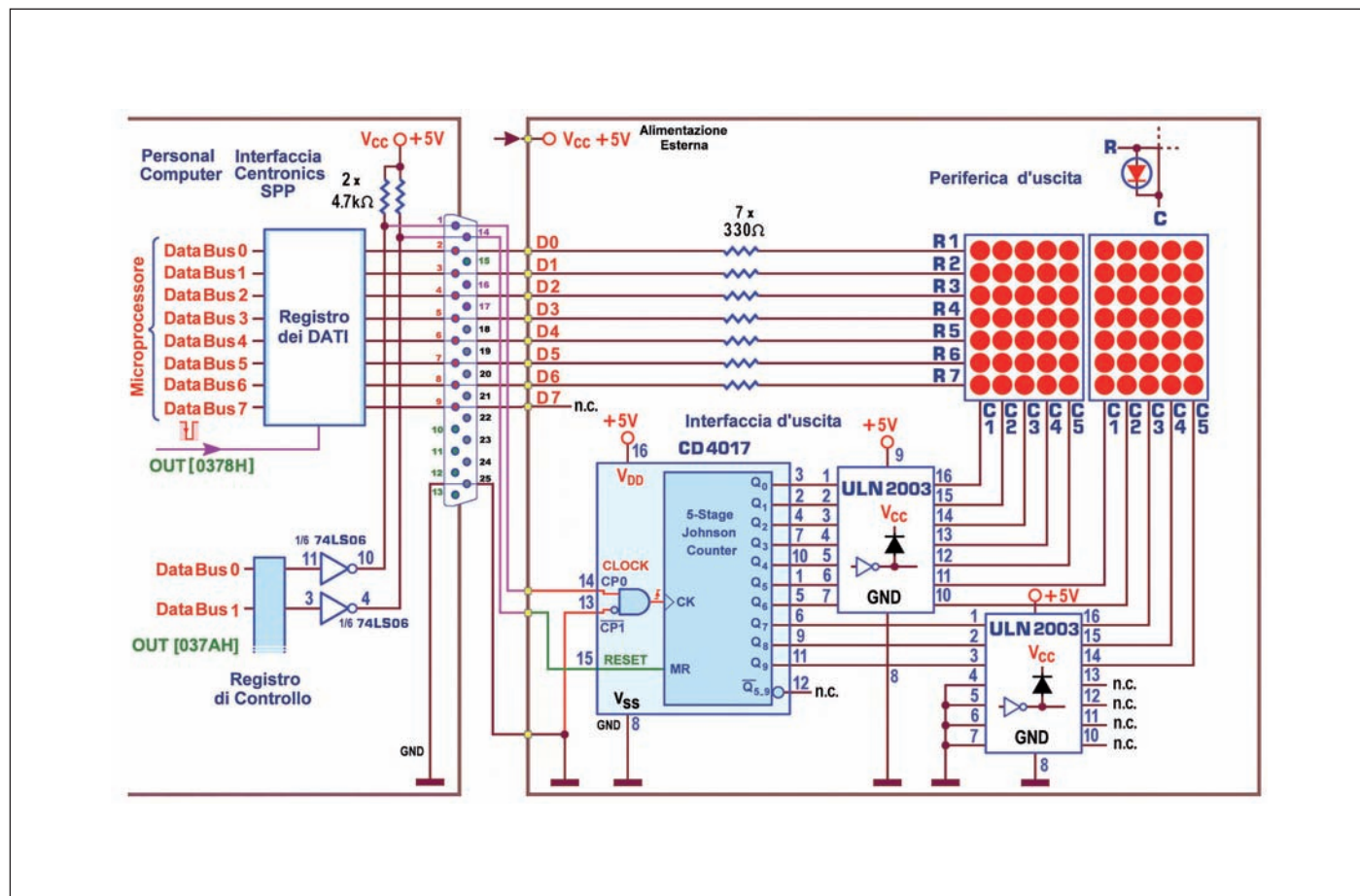



Figura 8 - Dot Matrix 7x5 LED Display: controllo con 5-stage Johnson Decade Counter 4017 tramite Porta Parallela SPP

Realizzare un prototipo SU PIASTRA MILLEFORI

Qualche pagina per (tentare) di spiegare come realizzare un prototipo su millefiori. L'operazione è un processo mentale che si acquisisce con il tempo, dunque al primo tentativo non aspettiamoci miracoli..

L'impiego è esclusivamente per componenti discreti normali, non a montaggio superficiale. In realtà esistono piastre per prototipi che prevedono l'uso di componenti SMD, ma si tratta di applicazioni specifiche, sicuramente non alla portata dell'hobbista ad inizio carriera.
Come è facile dedurre si tratta di un circuito stampato che riporta solamente piazzole, in genere in formato eurocard (100 x 160 mm) ha piazzole disposte lungo i due assi a distanza regolare "standard" di 2.54 mm (1/10 di pollice). Sono reperibili senza difficoltà presso i rivenditori di materiale elettronico e sono disponibili in bakelite e in vetronite, monofaccia o doppia faccia, anche con fori metallizzati, ovvero con il collegamento elettrico interno al foro tra le due piazzole opposte. La piastra millefori è utilizzabile in due modi, il primo riprende l'uso del circuito stampato classico e sfrutta le piazzole, unite da sottile filo per cablaggi (o da una semplice goccia di stagno), per realizzare le piste. Il metodo è valido per circuiti con una medio – bassa densità di componenti e per circuiti non troppo

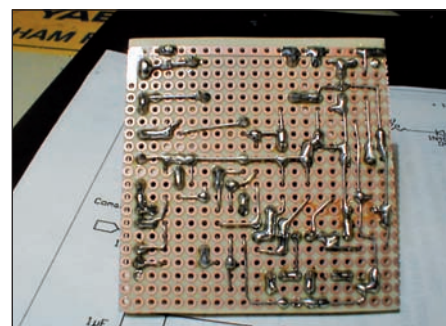



Figura 1: Cablaggio con i soli reofori

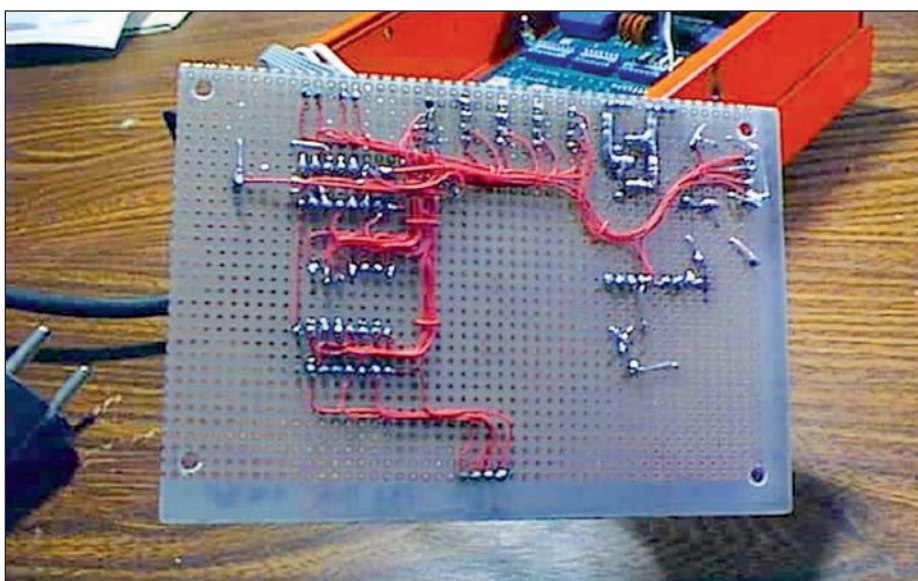



Figura 2: Cablaggio interamente filato

di DANIELE CAPPA

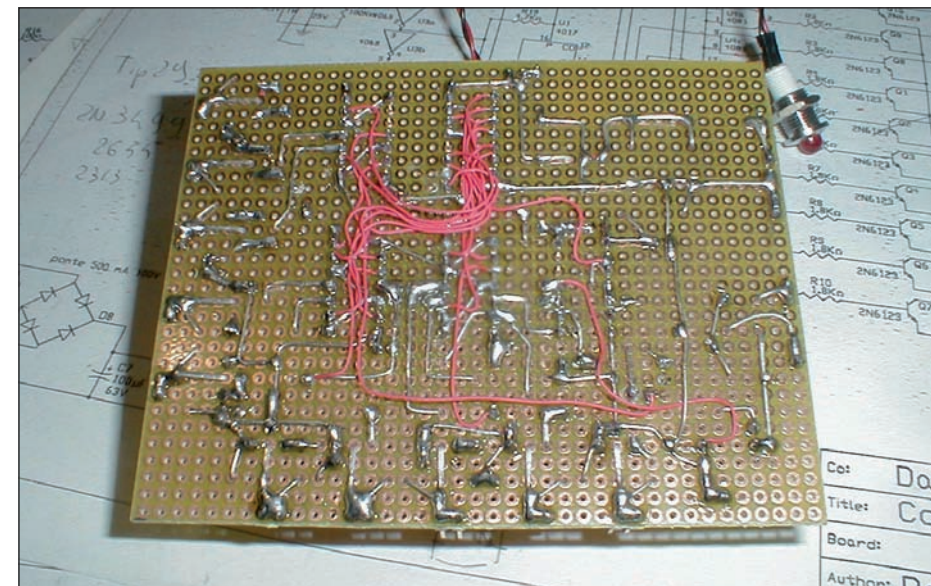



Figura 3: Il tester prova ascensora, lato saldature

complessi. Si presta, seppur con qualche limite, anche per l'uso in radiofrequenza. Il risultato estetico e la stabilità meccanica sono buone, se si è lavorato con ordine è del tutto paragonabile a un circuito stampato classico.

La piastra millefori esprime meglio le sue caratteristiche nell'uso in circuiti digitali ad alta densità di componenti discreti (e di collegamenti). Qui i collegamenti sono realizzati esclusivamente utilizzando filo da cablaggi molto fine (0.25 mm) rivestito in teflon, non pensiamo neppure di realizzare qualcosa di compatto utilizzando il filo telefonico citato più avanti, il risultato sarebbe orrendo e di infima affidabilità.

Il montaggio va realizzato con ordine e con molta attenzione, dimenticare, o peggio sbagliare un collegamento, può significare la perdita di molte ore. L'eventuale errore è più probabile sia rintracciabile strumentalmente che ricontrollando il circuito con la lente… in queste condizioni effettuare modifiche successive al circuito è possibile, ma estremamente difficoltoso.

Il sistema è adatto a chi abbia una ottima vista e la necessaria esperienza, decisamente sconsigliato a un principiante!

Il risultato è esteticamente meno gradevole, almeno guardandolo dal lato saldature, anche se è possibile apprezzare un prototipo portato a termine con cura e ordine; non è assolutamente adatto a montaggi a radiofrequenza, la stabilità meccanica è buona, sempre che tutto sia stato ben realizzato.

HO LO SCHEMA ELETTRICO, COME PASSO DALLA CARTA ALLA REALTÀ DEL PROTOTIPO?

Con calma e per gradi, per ogni cosa è necessaria una buona dose di esperienza e questa è da costruire sulla propria pelle, nessuno potrà mai regalarvela. Verranno citati degli esempi senza entrare troppo nel dettaglio, con il proposito (quasi sicuramente disilluso) di insegnare un metodo, non copiare lo schema.

Osserviamo lo schema elettrico e localizziamo le "zone".

- L'alimentazione con i suoi stabilizzatori e quanto gli sta intorno
- Gli ingressi con eventuali filtri
- Il cuore del circuito, quello che svolge la funzione richiesta
- I finali, gli attuatori di uscita, i relè, o quanto è dal lato opposto dell'ingresso

Lo schema elettrico andrà stampato e tenuto a portata di mano, teniamo a portata di mano una matita e una gomma adatta. In questo caso i nostri beneamati CAD per circuiti stampati non servono a

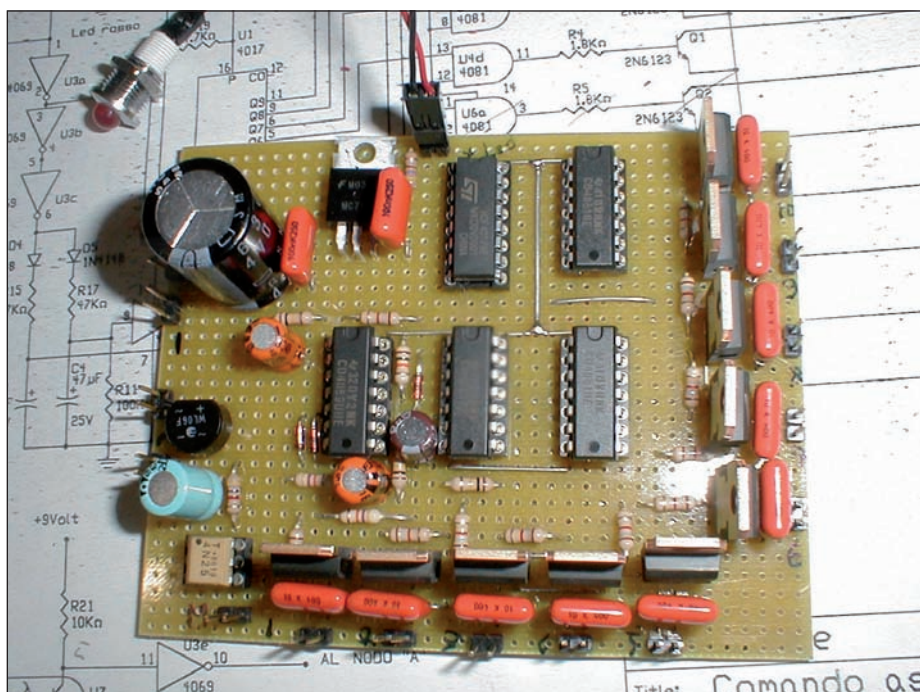



Figura 4: Il tester prova ascensore, lato componenti

nulla, a meno che non desideriamo romperci la testa realizzando a mano, senza ago, ma con il filo, lo stampato che ha progettato il PC. Vi garantisco che è tutt'altro che semplice, anche per circuiti poco complessi.
Abbiamo fatto la spesa dobbiamo decidere quanto spazio è necessario al nostro montaggio, a volte avremo degli spazi obbligati, altre volte no, Contenitore alla mano decidiamo lo spazio che abbiamo a disposizione, verifichiamo se sulla basetta deve starci tutta l'elettronica o se parte è da montare esterna. Finali su aletta, trasformatori di alimentazione di solito sono montati esterni alla piastra che supporta il cuore del nostro circuito.
Dividiamo ora la basetta in zone, così come le abbiamo localizzate sullo schema elettrico.
Seguiamo sempre un ordine preciso, ad esempio gli ingressi a sinistra e le uscite a destra, l'alimentazione in alto, o in basso. Se la cosa ci è di aiuto con carta & matita riportiamo gli ingombri dei componenti più grandi, oppure riportiamoli su carta, per rendere visivamente più chiara la situazione.
Localizziamo i componenti che hanno più collegamenti, dunque chip in genere, se di questi alcuni hanno dei collegamenti comuni provvederemo a metterli vicini curando che l'orientamento e la posizione faciliti queste connessioni. In questa fase valutiamo le connessioni ripetute, ovvero quei collegamenti che si ripetono spesso, resistenze di pullup sulle logiche (che potremmo probabilmente sostituire con un array di resistenze con un capo in comune), le uscite verso gli attuatori di uscita (le resistenze di base dei transistor comandati da logiche o microcontrollori). Questi sono esempi in cui molti elementi sono da trattare come se si trattasse di un elemento singolo, anche se moderatamente divisibile...
Raggruppare insieme le resistenze di base e i transistor che comandano (per esempio) alcuni relè, collocare il diodo di protezione nelle immediate vicinanze della bobina del relè, rende la piastra più "comprensibile". In seguito, quando si tratterà di intervenire sull'opera compiuta, la cosa potrebbe essere di aiuto.
Queste operazioni con il tempo si riveleranno delle "astuzie", ovvero... da un chip escono 8 bit che vanno ad altrettante resistenze (cosa c'è dopo per ora non ci importa), collocare le otto resistenze in fila davanti a ogni pin del chip interessato crea un "corridoio", dal lato saldature evidentemente, in cui è possibile far pas-

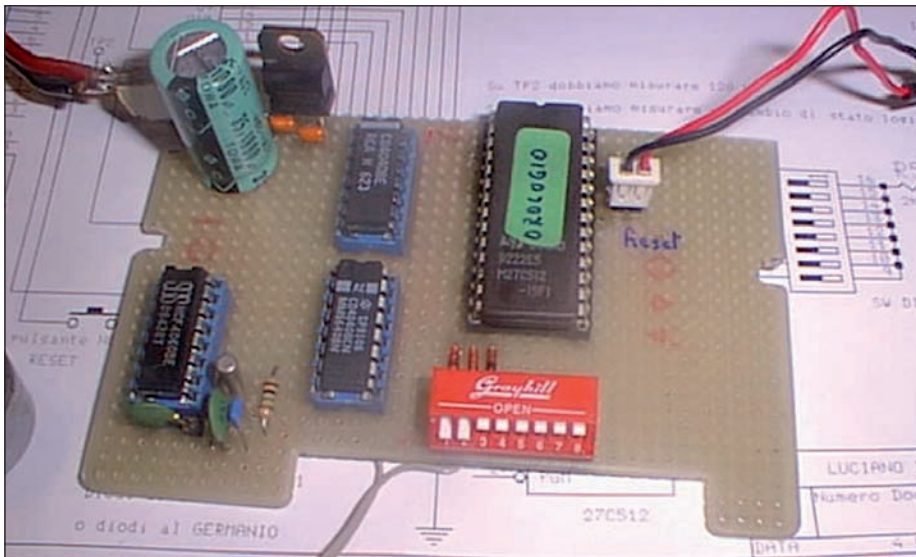



Figura 5: Lato componenti del comando della campana

sare altri collegamenti. In questo caso la cosa è da sfruttare evitando altre soluzioni più compatte, ma più invasive sulla disposizione dei componenti, quali potrebbero essere le medesime otto resistenze montate in verticale. In quest'ottica se abbiamo spazio a disposizione qualsiasi componente sarà montato occupando per intero lo spazio che gli è necessario. In quest'ottica realizzeremo solamente le saldature necessarie, ovvero se un reoforo della nostra resistenza è saldato sul pin del chip e l'altro sulla base del TR effettueremo questa saldatura solamente al momento in cui inseriremo quest'ultimo. Un reoforo non saldato è visibile senza problemi, se lo saldiamo alla piazzola senza effettuare alcun collegamento potremmo… lascialo indietro, ovvero dimenticarlo per strada.
Spesso le alimentazioni e gli stabilizzatori richiedono delle capacità verso massa, un piccolo condensatore ceramico multistrato da 10 - 100 nF, collegheremo anche loro, nel frattempo iniziamo a delineare i passaggi delle due alimentazioni, positivo e massa, che per ora non colleghiamo ancora.

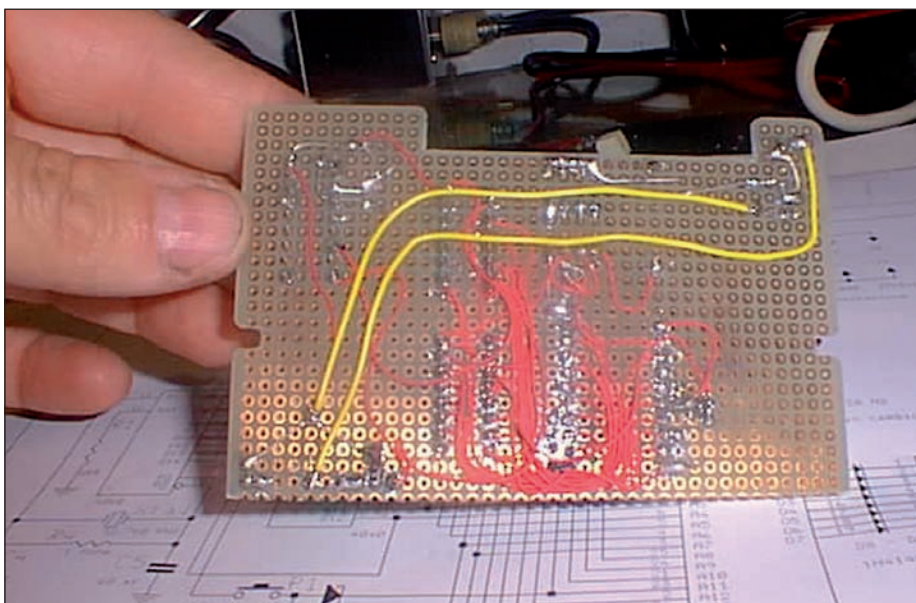



Figura 6: Lato saldature, i due fili gialli sono l'alimentazione di una campana esterna

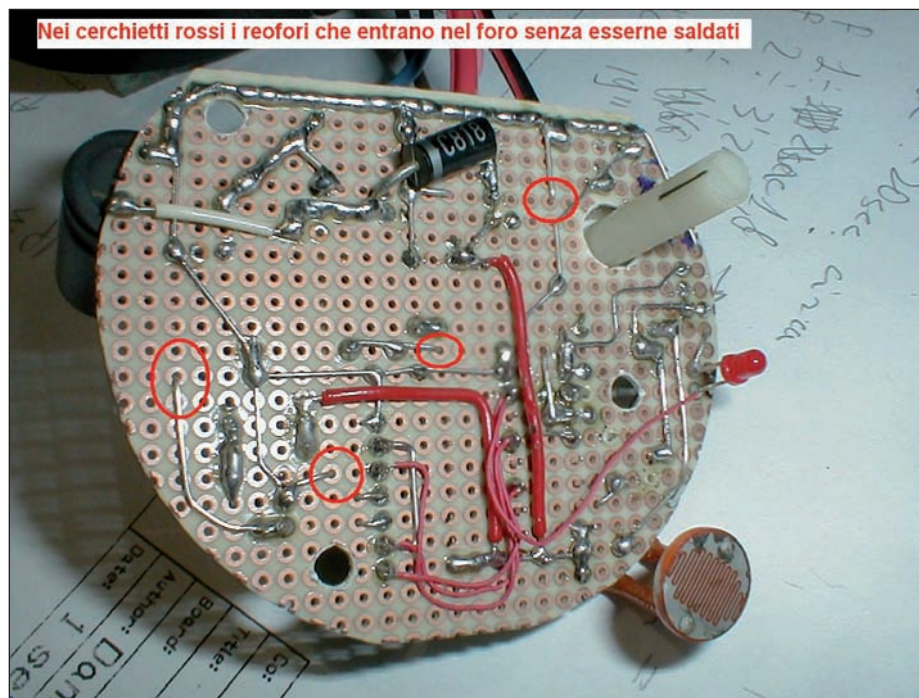



Figura 7: Lato saldature, Nei cerchietti i reofori non saldati sul passaggio nel foro

Posizioniamo quindi i componenti interessati, saldiamo gli zoccoli degli integrati e passiamo ai componenti ripetuti, li saldiamo e li colleghiamo, probabilmente sarà sufficiente una goccia di stagno tra due piazzole adiacenti. I collegamenti appena più lunghi saranno realizzati con il reoforo del componente opportunamente sagomato. Una precauzione però... se riteniamo che in fase di collaudo un componente possa essere sostituito è una buona idea effettuare le sue saldature lontane dalla piazzole in cui passano i reofori, questi ci permetterà di rimuoverlo semplicemente tagliando il reoforo, senza neppure ricorrere al dissaldatore.
Non dimentichiamoci mai di spuntare sullo schema cartaceo il componente che abbiamo appena montato, quando abbiamo effettuato un collegamento provvediamo a segnarlo sulla carta. Dimenticare un componente o un collegamento è un attimo, il tempo perso per rintracciarlo solamente dopo due giorni è enorme.
Con questo metodo ci liberiamo di una buona parte del nostro circuito. Passiamo quindi ai componenti minori, quelli la cui configurazione non è ripetuta più volte, la resistenza di base del singolo transistor (lui compreso), la rete RC sul pin del chip. Anche qui tratteremo ogni singolo gruppetto in modo autonomo, in modo da non dimenticarne un pezzetto per strada... Se sul nostro chip c'è un ritardo formato da una resistenza e un piccolo elettrolitico, dobbiamo avere l'accortezza di posizionarli e collegarli entrambi. Il transistor andrà posizionato e con lui i componenti che gli stanno intorno. La cosa genera meno confusione dato che inevitabilmente alcuni collegamenti andranno effettuati prima e altri dopo. In questo frangente avremo sicuramente a che fare con alcuni collegamenti di massa, dove possibile provvederemo a collegarli tra loro, magari includendo il corrispondente pin dell'integrato.
Pian piano ci avviciniamo all'uscita e il nostro foglio di carta è ormai pieno di tratti a matita, malgrado questo non abbiamo ancora collegato nessun filo da cablaggi, i collegamenti che devono attraversare tutta la basetta sono ancora da realizzare. Lasciamoli dove sono e passiamo ai...

... COLLEGAMENTI DI MASSA E DELL'ALIMENTAZIONE

Saranno gli ultimi ad essere collegati, dove possibile utilizzeremo i punti già stabiliti posizionando i componenti. Qui abbiamo più possibilità, possiamo sfrut-

tare un collegamento centrale per il positivo e un semianello esterno per la massa (prudenza vuole di non chiudere mai la massa ad anello, ovvero non fare mai in modo che un filo di massa percorra tutto il perimetro della piastra). Oppure la massa in basso e il positivo in alto, basta essere coerenti e nel caso di confusione differenziare la cosa utilizzando sezioni di guaina isolante di colore diverso, anche se non è necessaria una protezione sul filo.
In queste condizioni possiamo impiegare filo di rame nudo, o stagnato, dal diametro modesto, ma non particolarmente fine, anche il citato filo telefonico (0,5 mm di diametro), avendo cura di non utilizzare la parte che abbiamo spelato con le forbici. Queste lasciano un segno netto e profondo nel rame tenero e quasi sicuramente in quel punto il filo si romperà. Lo farà nel momento peggiore, mantenendosi accuratamente nascosto. Meglio non cercar rogne… speliamo il filo a tratti di una ventina di centimetri per volta, tendiamolo bene utilizzando due pinze ed eliminiamo i centimetri alle due estremità, quelli che abbiamo manipolato con forbici e pinze.
Uno dei due collegamenti può essere in parte effettuato dal lato componenti, inserendo dei ponticelli ad arte, oppure realizzando una ordinata ragnatela di filo nudo. Se la corrente richiesta è più alta i collegamenti delle alimentazioni potranno essere realizzati con due fili vicini che provvederemo unire con un cordoncino di stagno per tutta la loro lunghezza, non è bellissimo, ma un paio di ampere ci possono passare.
Abbiamo ormai collocato tutti componenti, abbiamo realizzato le connessioni più facili e più corte, le alimentazioni sono a posto, mancano solamente i collegamenti più impegnativi, le connessioni alle logiche fisicamente non contigue, bus indirizzi che da un lato abbiamo collegato con ordine, ma dall'altro non gli corrisponde la sequenza analoga. Su un cìrcuito stampato avremmo avuto un mare di ponticelli o una doppia faccia, qui ci accontentiamo di realizzarli con filo da cablaggi fine (i citati 0,25 mm, rigorosamente monofilo), isolato in teflon. Non possiamo utilizzare nulla di diverso, un isolante in plastica si fonde con risultati penosi, un filo più grande ci impedisce di sagomarlo e farne delle matassine dalle dimensioni ragionevoli. Qui il filo del telefono non possiamo proprio usarlo!!
Anche qui a ogni collegamento corrisponde il corrispondente "spunto" sul foglio di carta, e anche in questo caso effettueremo le connessioni a gruppi, ovvero tutti gli indirizzi di un contatore piuttosto che tutte le uscite verso il mondo esterno

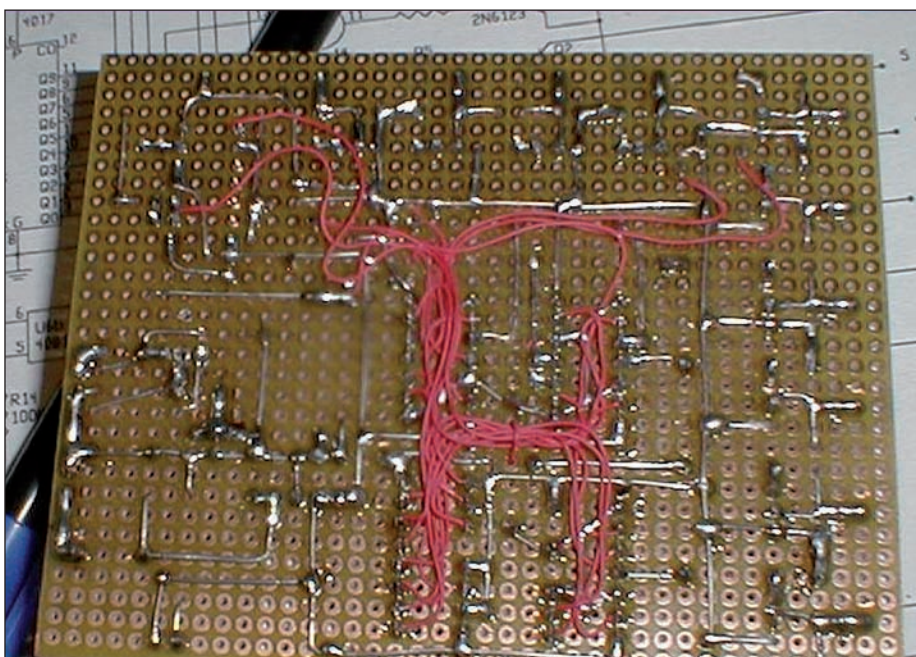



Figura 8: Millefori cablata con entrambi i metodi

o i collegamenti del display al microcontrollore. Faremo passare i fili tutti più o meno insieme, se questi sono molti possiamo "legarli" a matassina utilizzando lo stesso filo da cablaggi che, passando in due fori della piastrina, sarà fermato dal lato componenti con un "poco elettronico nodo..." Dopo qualche sera siamo venuti a capo del nostro prototipo, se abbiamo lavorato bene, con ordine, questo funzionerà secondo le previsioni, o meglio secondo quanto dichiarato dello schema elettrico. Abbiamo però alcune considerazioni da fare.

CIRCUITI DIGITALI, BASSA FREQUENZA E RADIOFREQUENZA

Il sistema descritto è adatto a piastre digitali, dove i segnali non sono velocissimi, non ci sono grosse correnti in gioco, i due livelli logici sono ben definiti e a bassa impedenza...

Il mondo reale non è sempre così, in un circuito di un amplificatore audio potrebbero essere necessarie alcune precauzioni.

Partiamo dal supporto, la bakelite che io impiego con tanta allegria, è poco adatta impieghi seri, dunque nel caso di un montaggio a cui tenete particolarmente andare su una bella piastrina di vetronite, probabilmente le caratteristiche tra i due supporti non hanno alcuna importanze nel vostro montaggio, ma l'aspetto finale è più professionale... se il tutto è un circuiti RF allora nessun dubbio, vetronite senza discutere.

Le connessioni realizzate con filo da cablaggi sono inevitabimente lunghe, parallele tra loro e realizzate con fili molto fini, di conseguenza hanno una elevata capacità tra un collegamento e il suo vicino, cosa che non li rende adatti in circuiti RF, hanno un'alta resistenza dettata dal filo molto fine, dunque ci possono passare pochi mA.

I circuiti RF sono delle brutte bestie, e su millefori la cosa si fa più evidente. Nel caso la nostra realizzazione rientri in questa categoria è bene rispettare alcune regole.

..chi predica bene razzola male, nella foto il PTO già pubblicato su questa testata (rivista n. 274) e realizzato su millefori in bakelite, a mia discolpa la bassa frequenza di funzionamento, poco più di 4 MHz

- i singoli stadi devo essere separati, talvolta schermati tra loro, anche utilizzando della normale carta di spagna (si tratta di fogli di rame o ottone sottilissimi, 1/10 mm, si taglia con le forbici e si acquista in negozi di modellismo)
- le alimentazioni saranno tutte bypassate (orrendo) da dei piccoli condensatori ceramici
- eventuali bobine non dovranno guardarsi tra loro, ovvero andranno montate a 90 gradi una dall'altra
- i collegamenti vanno effettuati più corti possibile, piuttosto rinunciamo all'estetica e al montaggio dei componenti sui due assi... brutto, ma a volte utile
- ovviamente impiegheremo solamente supporti in vetronite

In queste condizioni possiamo cimentarci sino alla soglia delle VHF, anche un pelo oltre, se siamo bravi...

Siamo alla fine del nostro discorso, consapevoli che insegnare un metodo è difficile quanto insegnare ad andare in bicicletta, si possono dare consigli, linee guida, ma l'unica vera maestra è anche in questo caso l'esperienza sulla propria pelle. Dunque rassegniamoci fare qualche brutta figura, magari esponendoci alle critiche che certo su quest'argomento non mancano. Personalmente ho rinunciato al circuito stampato (se non in casi di assoluta necessità) molti anni fa. Un progetto è tipicamente un solo esemplare e la millefori si presta ottimamente allo scopo.

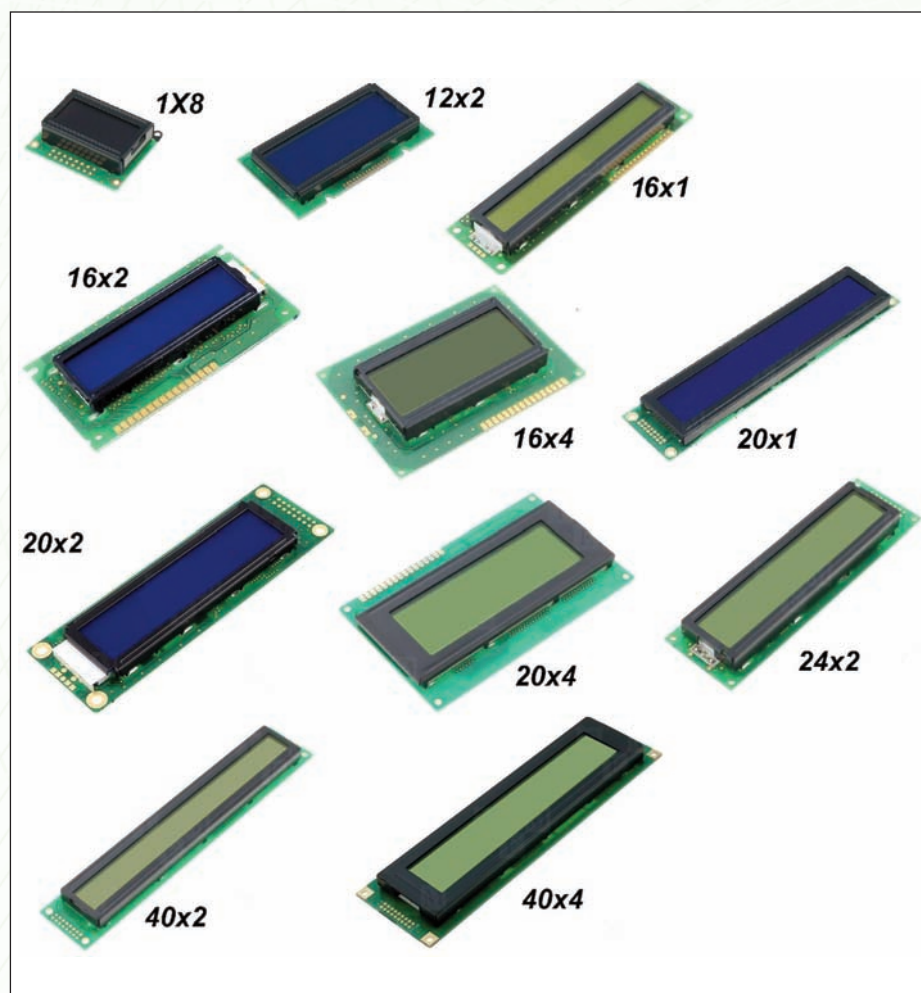
L'impegno mentale necessario alla progettazione "dal vivo" della nostra piastrina è un ottimo esercizio per progettare (a mano) il corrispondente stampato, se mai vedrà la luce. Le foto sono tutte autoprodotte, molte sono già passate su queste pagine accompagnando il progetto originale. ❑

CODICE MIP 2834998

Corso di programmazione Mikropascal per PIC
(parte sesta)

LCD ALFANUMERICI
E STRUMENTI DI SVILUPPO *per PIC* parte 2

Nella prima parte della puntata concluderemo l'argomento dei display LCD alfanumerici parlando di LCD di formato differente dal 2x16, dell'ASCII Chart e del tool Lcd Custom Character Editor integrato nell'IDE MikroPascal Pro for PIC. Dedicheremo quindi la seconda parte ad una panoramica su kit di sviluppo, demo-board, programmatori e strumenti di simulazione software per controllori PIC. Il lettore, alla luce di quanto esposto, potrà valutare direttamente l'opportunità di dotarsi di qualcuno di questi strumenti al fine di affrontare con maggiore efficacia le successive parti del corso.

Figura 1- I formati di LCD alfanumerici standard Hitachi HD44780 più comuni

Nella precedente puntata del corso abbiamo imparato a gestire i display LCD alfanumerici (standard Hitachi HD44780) con i controllori PIC utilizzando codice *MikroPascal Pro for PIC*. In particolare, abbiamo sia discusso l'architettura generale ed il funzionamento di un LCD di questo tipo, sia parlato delle routine di libreria MikroPascal e dei relativi comandi predefiniti che è possibile utilizzare per il controllo del display. In particolare, abbiamo proposto delle dimostrazioni pratiche sull'argomento che coprono sia aspetti hardware che firmware incentrate soprattutto sull'impiego di display LCD *2x16* (2 righe di sedici caratteri ciascuna). Quanto esposto può essere facilmente esteso a display LCD dello stesso tipo ma di diverso formato (*Figura 1*).
Quanto acquisito ci mette nelle condizioni di poter utilizzare un display LCD alfanumerico per la comunicazione tra un generico processo a microcontrollore impiegante un PIC e l'utente che si interfaccia a quel processo.
In questa puntata concluderemo l'argomento parlando di *ASCII Chart* e mo-

di ANTONIO GIANNICO

Figura 2-Lcd_Parte2_Esempio1. Quanto detto e mostrato per i display alfanumerici, standard Hitachi HD44780 compatibile 2x16, nella scorsa lezione, vale senza sostanziali differenze per display 1x16, 2x20, 4x16, 4x20, 2x40 e 2x24. Si analizzi a tale proposito questo progetto di esempio (Project Manager riportato in alto a destra, listato del codice, commento in testa allo stesso e il risultato della visualizzazione su display di diverso formato).

strando l'impiego del tool *Lcd Custom Character Editor,* entrambi integrati all'interno dell'IDE MikroPascal Pro for PIC. In particolare, vedremo come questi due tool possano risultare estremamente utili durante la stesura di codice MikroPascal finalizzato alla gestione degli LCD alfanumerici.

Come osservato alla fine della precedente puntata, giunti a questo punto, lo studio degli argomenti potrebbe scoraggiare coloro che, avendo poca dimestichezza con il mondo dei controllori o dell'elettronica in genere, potrebbero trovare la sperimentazione pratica degli stessi argomenti difficoltosa a causa della mancanza di strumenti adeguati che non possono a questo punto limitarsi al solo IDE MikroPascal. Più si proseguirà nello sviluppo del corso è più si renderà necessario, per un'efficace acquisizione dei concetti, disporre di alcuni strumenti ulteriori, non necessariamente costosi o di difficile reperibilità. Comprendendo la naturale incertezza a compiere questo passo da parte dei lettori meno esperti e nello spirito che ci deve contraddistinguere di sperimentare praticamente per imparare, dedicheremo la seconda parte della puntata ad una panoramica sui *tool di sviluppo* per microcontrollori PIC, sulle *demo-board* più diffuse sul mercato e sui *programmatori* per controllori PIC, soprattutto di tipo *entry-level*, oltre che a strumenti virtuali di simulazione. L'argomento, ci rendiamo conto, esula in qualche modo dal corso, incentrato soprattutto sulla programmazione MikroPascal per PIC, tuttavia ne è complementare ed è di importanza fondamentale se si vuole colmare efficacemente il gap che esiste tra teoria e pratica.

Il lettore che ancora non abbia provveduto autonomamente in tal senso, alla luce delle nozioni acquisite, avrà la possibilità di valutare, con cognizione di causa, l'opportunità o meno di dotarsi di qualcuno di questi strumenti al fine di affrontare le successive parti del corso in maniera più divertente ed efficace.

L'argomento è volutamente trattato in maniera non eccessivamente ampia ma approfondita quanto basta per dare l'opportunità al lettore di cercare, confrontare, valutare gli strumenti che fanno maggiormente al caso proprio, prima di acquistarne eventualmente alcuni, mantenendo un giusto rapporto tra utilità e costo.

Il fatto di non vincolarci necessariamente a uno specifico kit ci consente di rendere il corso più generale e flessibile ma non per questo meno approfondito, evitando che lo sperimentatore debba dotarsi necessariamente, da un punto di vista hardware, di un dato strumento piuttosto che di un altro.

TUTTI I FORMATI PIÙ COMUNI DI LCD ALFANUMERICI HITACHI HD44780 COMPATIBILI

Quanto detto e mostrato per LCD alfanumerici standard Hitachi HD44780 compatibili 2x16 nella scorsa lezione vale, senza sostanziali differenze, per i display dei formati riportati in Figura 1.

Ci limitiamo a riguardo a mostrare l'esempio firmware di Figura 2 la cui esecuzione è mostrata rispettivamente per LCD 2x16, 2x20, 4x16 e 4x20 (Figura 2).

Tralasciamo di riportare lo schema elettrico dell'applicazione rimandando, a quelli riportati negli esempi della precedente puntata n.5 di cui la presente può essere considerata a tutti gli effetti una continuazione.

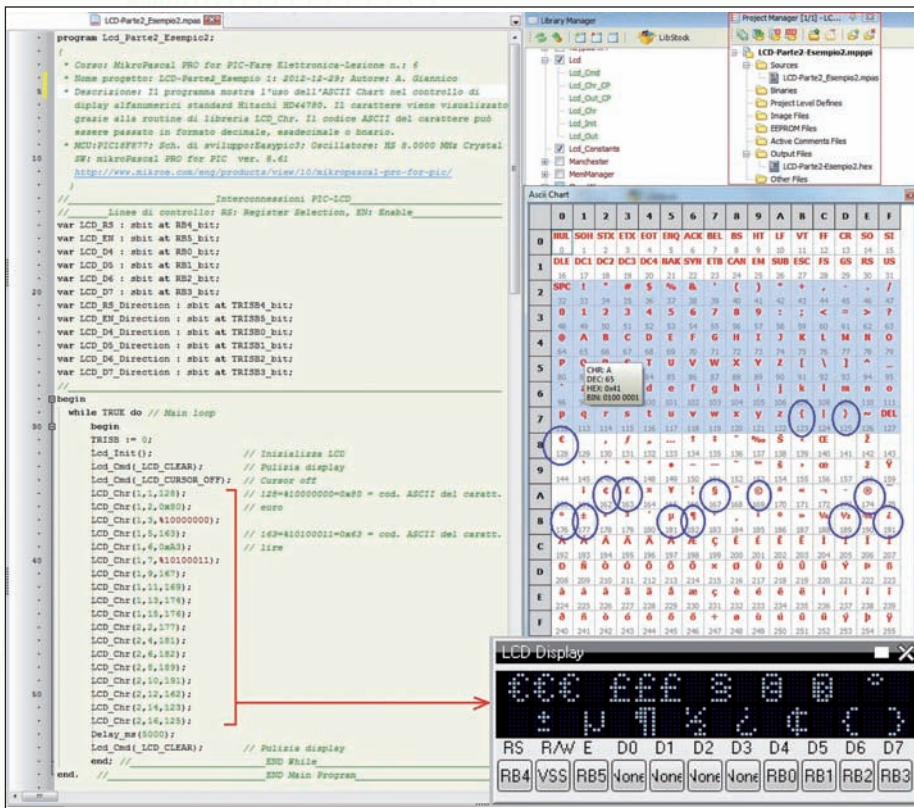



Figura 3-Lcd_Parte2_Esempio 2. Attraverso l'ASCII Chart [2], accessibile attraverso ToolsàASCII Chart o in alternativa attraverso l'apposita icona "View ASCII Chart", è possibile visualizzare la codifica dei caratteri ASCII. I codici possono essere utilizzati (in formato esadecimale, decimale o binario), per esempio come argomento della routine Lcd_Chr, per effettuare la rappresentazione dei rispettivi caratteri sul display).

ASCII CHART

L'*ASCII Chart* è un tool integrato nell'IDE MikroPascal molto utile quando si ha a che fare con la gestione di LCD alfanumerici. Esso è accessibile dal menu Tools (*Tools è ASCII chart*) o in alternativa dall'apposita icona *"View ASCII Chart",* riporta, per ciascun carattere, il corrispondente codice ASCII nei tre formati rispettivamente *esadecimale, decimale e binario* ed è di ausilio alla stesura di codice per gestione di display LCD alfanumerici, come l'esempio di Figura 3 (Esempio2), cui si rimanda, mostra chiaramente.

LCD CUSTOM CHARACTER EDITOR

MikroPascal PRO for PIC integra al suo interno anche il Tool *Lcd Custom Character Editor,* accessibile attraverso il menu *ToolsàLcd Custom Character*. Esso consente, definito il carattere personalizzato disegnato sulla matrice di punti 5x7 o 5x10, di generare il codice di una procedura in grado di far comparire sul display il carattere stesso. Il carattere personalizzato viene definito selezionando sulla rappresentazione della matrice 5x7 o 5x10 i pixel che si desidera "accendere" (Figura 4). Prima di avviare la generazione del codice (pulsante "*Generate Code*"*-Figura 4*) è necessario selezionare il compilatore dal momento che lo stesso tool è indistintamente integrato negli ambienti di sviluppo MikroC, MikroPascal e MikroBasic PRO for PIC). A tale proposito è utile analizzare con attenzione il codice MikroPascal dell'Esempio3 riportato in Figura 4. Nella trattazione proposta nella precedente puntata ci siamo riferiti soprattutto a display LCD con caratteri a matrice di punti 5x7; come è possibile osservare dall'analisi di questo tool (si osservi la selezione *Font Size-Figura 4*) esistono in commercio e sono abbastanza diffusi anche display alfanumerici a matrice di punti 5x10 del tutto simili a quelli 5x7 cui abbiamo fatto riferimento con gli esempi della precedente lezione e della presente. Concludiamo la serie di esempi di codici MikroPascal Pro for PIC per il controllo di LCD alfanumerici ponendo all'attenzione del lettore il codice di Figura 5 (Lcd_Parte2_Esempio4). Il codice esegue il test dell'LCD mostrando ciclicamente sullo stesso la sequenza di caratteri maiuscoli dell'alfabeto seguita dalla sequenza dei caratteri minuscoli. Per far ciò viene scandita la tabella dei codici ASCII (vedi Figura 3) escludendo dalla scansione i codici che non rappresentano caratteri alfabetici. Mentre la variabile m (di tipo Byte) scandisce i codici ASCII, l'indice j scandisce le righe del display e l'indice k le colonne.

KIT DI SVILUPPO, PROGRAMMATORI, DEMO-BOARD E STRUMENTI VIRTUALI DI SIMULAZIONE PER PIC

Come accennato nelle prime puntate, per poter editare e compilare il codice, trasferire il firmware nella memoria flash del microcontrollore, testare, modificare e ritestare i progetti firmware, è necessario dotarsi di strumenti sia software (essenzialmente l'ambiente di sviluppo *MikroPascal Pro for PIC*) che hardware.
Per quanto concerne gli strumenti hardware è possibile agire essenzialmente in due modi:

1. utilizzare *demo-board* e kit di sviluppo disponibili sul mercato;
2. sviluppare autonomamente il PCB per lo specifico circuito eseguendo il montaggio dei componenti sullo stesso al fine di disporre della board finale.

Nel primo caso, lo strumento integra spesso on-board il circuito programmatore che interfacciato, generalmente via USB al proprio PC, consente il trasferimento del firmware nella memoria flash del PIC. Nel caso in cui si tratti di una demo-board che non integra il circuito programmatore è possibile utilizzare un programmatore esterno, da collegare, generalmente via USB, al proprio PC e attraverso un opportuno cavetto segnali alla demo board.
Anche nel secondo caso è generalmente necessario prevedere un programmatore esterno da collegare al PC da un lato e alla board dall'altro, ricadendo di fatto nella situazione precedente, a meno di non integrare sulla board anche il circuito programmatore. In ogni caso, il vantaggio è quello di poter programmare il control-

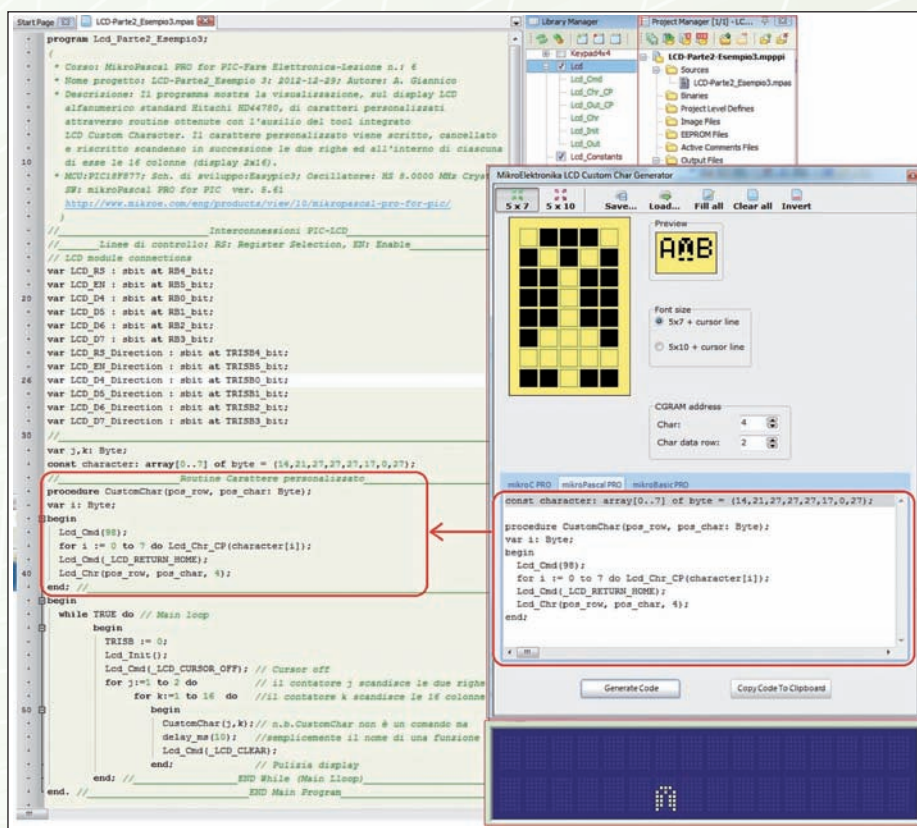



Figura 4-Lcd_Parte2_Esempio3. Il Tool Lcd Custom Character Editor [2] accessibile attraverso ToolsàLcd Custom Character) consente, definito il carattere personalizzato disegnato sulla matrice di punti 5x7 o 5x10, di generare il codice di una procedura in grado di far comparire sul display il carattere stesso. Il carattere personalizzato viene definito selezionando sulla matrice 5x7 o 5x10 i pixel che si desidera "accendere".

lore ripetutamente senza rimuoverlo dalla scheda sulla quale è montato. In altre parole è possibile eseguire la cosiddetta *programmazione in-circuit*.

In alternativa, è possibile utilizzare, per la sola operazione di programmazione, schede che integrano solo questa funzione e che prendono il nome di programmatori. In questo caso il controllore viene posto su un apposito zoccolo del programmatore, programmato con il firmware, quindi rimosso e montato sulla scheda finale che implementa in hardware l'applicazione. Come è facile comprendere, questo modo alternativo di agire è piuttosto scomodo soprattutto in fase di sviluppo e test del firmware in quanto può costringere a rimuovere il controllore dalla sua board più volte al fine di sottoporlo a riprogrammazione. Per questo motivo si tratta di un modo di procedere adatto al più in fase di produzione piuttosto che a quella di sviluppo e prototipazione.

La procedura al punto 2 necessita inoltre di maggiore esperienza tecnico-pratica comportando la necessità di controllare interamente la realizzazione dell'applicazione sia negli aspetti hardware che firmware. L'impiego di kit di sviluppo HW/SW o di demo-board consente invece di eliminare quasi del tutto le incertezze e le problematiche legate all'hardware e di concentrare maggiormente l'attenzione sullo sviluppo firmware dell'applicazione. Da un punto di vista del test del firmware è invece possibile agire in due modi:

1. utilizzare *simulatori e debugger software* integrati nell'ambiente di sviluppo;
2. utilizzare *simulatori software* esterni e a se stanti che simulano l'hardware di una specifica scheda di sviluppo o un hardware generico (che è possibile entro certi limiti configurare secondo le proprie specifiche esigenze) e l'esecuzione su questo hardware del firmware .hex.

Quanto riportato al punto 1 è un argomento specifico strettamente legato, nel nostro caso, all'impiego dell'IDE MikroPascal la cui trattazione viene rimandata a puntate successive; quanto descritto invece al punto 2 è oggetto dei paragrafi che seguono.

PROGRAMMATORI PER SVILUPPO E PER PRODUZIONE

Un programmatore è un dispositivo che consente fisicamente il trasferimento del firmware (file .hex) nella memoria flash del microcontrollore. Per i controllori PIC, ne esistono in commercio sia prodotti da Microchip che da altre case costruttrici. Sono dotati di uno zoccolo sul quale viene fissato il controllore per il trasferimento del firmware. A seconda del modello, consentono la programmazione di soli controllori PIC di Microchip, di una o più famiglie, oppure di controllori di diverse famiglie ed eventualmente non solo Microchip. In questo ultimo caso si parla comunemente di programmatori universali. Alcuni sono dotati di zoccoli multipli al fine di consentire la programmazione di diversi modelli di controllori oppure la programmazione di più controllori dello stesso modello simultaneamente. Si tratta, per questo, di strumenti utili soprattutto in fase di produzione più che in fase di sviluppo e test dal momento che la programmazione consentita non è di tipo in-circuit. Una ulteriore limitazione è legata al fatto che diventa scomodo programmare con questo tipo di programmatori controllori con package SMD, per esempio controllori in package QFP o PLCC. Volendo programmare in questo caso controllori aventi questo tipo di package è infatti necessario utilizzare degli adattatori, cioè dei piccoli PCB di adattamento che sul lato TOP riportano l'impronta del package SMD da saldare e sulla parte Bottom due file di pin a foro passante che riproducono il disegno di un package PDIP.

Quando si deve acquistare un programmatore è evidentemente necessario riferirsi a linee di prodotto differenti a seconda che esso debba essere utilizzato per sviluppo oppure per produzione. Nel nostro caso, dal momento che stiamo semplicemente portando avanti un percorso didattico, sarà opportuno rivolgere maggiormente l'attenzione alla prima serie di prodotti. Nell'ambito della progettazione e sviluppo e quindi in una certa misura anche in ambito didattico è inoltre più

```
program Lcd_Parte2_Esempio4;
{
 * Corso: MikroPascal PRO for PIC-Fare Elettronica-Lezione n.: 6
 * Nome progetto: LCD-Parte2_Esempio 4: 2012-12-29; Autore: A. Giannico
 * Descrizione: Il programma mostra sul display LCD alfanumerico standard
   Hitachi HD44780 la serie dei caratteri maiuscoli dell'alfabeto seguita
   dalla serie dei caratteri minuscoli. Il ciclo j scandisce le righe, il ciclo
   k le colonne del display ed il contatore m i codici ASCII dei caratteri da
   visualizzare. L'assegnazione del valore iniziale di m (65) ed i controlli if
   consentono di escludere dalla visualizzazione i caratteri non alfabetici.
 * MCU:PIC18F877; Sch. di sviluppo:Easypic3; Oscillatore: HS 8.0000 MHz Crystal
   SW: mikroPascal PRO for PIC  ver. 5.61
   http://www.mikroe.com/eng/products/view/10/mikropascal-pro-for-pic/
  }
//___________________________Interconnessioni PIC-LCD____________________________
//_______Linee di controllo: RS: Register Selection, EN: Enable__________________
// LCD module connections
var LCD_RS : sbit at RB4_bit;
var LCD_EN : sbit at RB5_bit;
var LCD_D4 : sbit at RB0_bit;
var LCD_D5 : sbit at RB1_bit;
var LCD_D6 : sbit at RB2_bit;
var LCD_D7 : sbit at RB3_bit;
var LCD_RS_Direction : sbit at TRISB4_bit;
var LCD_EN_Direction : sbit at TRISB5_bit;
var LCD_D4_Direction : sbit at TRISB0_bit;
var LCD_D5_Direction : sbit at TRISB1_bit;
var LCD_D6_Direction : sbit at TRISB2_bit;
var LCD_D7_Direction : sbit at TRISB3_bit;
//______________________________________________________________________________
var j,k,m: Byte;
begin
  m:=65;
  while TRUE do // Main loop
        begin
          TRISB := 0;
          Lcd_Init();                // Inizializzazione
          Lcd_Cmd(_LCD_CLEAR);       // Pulizia display
          Lcd_Cmd(_LCD_CURSOR_OFF);  // Cursor off
          for j:=1 to 2 do           // j scandisce le righe del display
                for k:=1 to 16  do   // k scandisce le colonne
                     begin
                       Lcd_Chr(j, k, m);
                       m:=m+1;// m scandisce i codici ASCII dei caratteri
                       if m=91 then m:=97;  // dell'alfabeto.
                       if m=123 then m:=65;
                       delay_ms(10);
                     end;
        end; //_____________________END While (Main Loop)_______________________
end. //____________________________END Main Program_____________________________
```

ABCDEFGHIJKLMNOP
QRSTUVWXYZabcdef

Figura 5-Lcd_Parte2_Esempio 4. Questo codice esegue il test dell'LCD mostrando ciclicamente sullo stesso la sequenza di caratteri maiuscoli dell'alfabeto seguita dalla sequenza dei caratteri minuscoli. Per far ciò viene scandita la tabella dei codici ASCII (vedi Figura 3) escludendo dalla scansione i codici che non rappresentano caratteri alfabetici.

importante la flessibilità del programmatore soprattutto nel poter consentire programmazione *in-circuit* e operazioni da *debugger*, diversamente, in ambito produzione è più importante la possibilità di poter programmare più dispositivi contemporaneamente e quella di poter programmare diversi tipi di dispositivi (EEPROM, FPGA, microcontrollori). Quest'ultima prerogativa è propria dei cosiddetti programmatori universali. Quelli più costosi e adatti alla produzione su larga scala consentono la cosiddetta *programmazione concorrente* (*gang*). In altri termini, si tratta di programmatori che consentono la scrittura simultanea ma sostanzialmente indipendente di più chip. Il programmatore si presenta agli occhi dell'utente come un'unità unica ma integra al suo interno a tutti gli effetti più programmatori indipendenti generalmente comunicanti con il PC attraverso un comune bus USB (hub USB integrato onboard). I *gang programmer* non vanno per questo confusi con programmatori dotati di particolari adattatori multisocket

Figura 6-Programmatore In-Circuit-Debugger PicFlash2 di Mikroelektronika [5]

che consentono la programmazione di più chip in parallelo. In questo caso infatti lo stesso segnale viene semplicemente ripartito su più chip, si degrada più facilmente ed in caso di errori non è possibile stabilire quale chip di quelli che si è cercato di programmare li abbia causati.
Volendo completare la panoramica sui programmatori da produzione occorre distinguere i programmatori standalone che possono essere utilizzati anche senza il PC e quelli non standalone che necessitano sempre di un PC per la loro gestione ed il trasferimento del firmware sul chip da programmare. Come già detto, in ambito di progettazione, verifica e prototipazione non sono necessari programmatori da produzione ma *programmatori in circuit/debugger e kit di sviluppo o demo board e/o evaluation board*. A questa linea di prodotti dedichiamo i paragrafi che seguono.

PROGRAMMATORI IN-CIRCUIT ED IN-CIRCUIT DEBUGGER (ICD)

Alcuni programmatori consentono la programmazione *in-circuit* del controllore. In questo caso è necessario predisporre, sulla board destinata ad ospitare il controllore ed a realizzare fisicamente l'hardware dell'applicazione, un connettore al quale connettere il programmatore. Quest'ultimo è dotato di un secondo connettore, generalmente USB per la connessione al PC. Il grande vantaggio di predisporre sulla board il connettore per la programmazione *in-circuit* e di utilizzare questo tipo di programmatori sta pertanto nella possibilità di riprogrammare più volte il controllore senza doverlo rimuovere dalla board stessa, cosa estremamente utile sia in fase di sviluppo, test e prototipazione che in occasione di aggiornamenti firmware dell'applicazione. Spesso si tratta di programmatori in-circuit che integrano anche funzioni da *debugger*. In questo caso si parla di *In-Circuit-Debugger* (*ICD*). La connessione USB tra programmatore e PC comporta l'installazione, su questo ultimo, di appositi driver mentre la connessione tra programmatore e scheda avviene generalmente attraverso un secondo cavo e quindi un connettore che riporta, oltre all'alimentazione, i segnali MCLR, PGC e PGD per la programmazione. Un esempio è costituito dal programmatore in-circuit-debugger *PicFlash2* di Mikroelektronika ([5], Figura 6).
La funzione di debugger integrata è a tutti gli effetti un'alternativa al debugger software, peraltro migliore dal momento che consente un debug in-circuit sull'hardware reale su cui si sta sviluppando l'applicazione. Questo consente l'analisi del corretto funzionamento del firmware dopo che lo stesso è stato fisicamente caricato nella flash del controllore PIC. Altri esempi sono costituiti dall'*ICD2* e *ICD3* di Microchip ([9], Figura 7) e dagli analoghi a basso costo *PicKit2 e PicKit3* (Figura 8), ancora di Microchip, che sono pensati per integrarsi con l'ambiente di sviluppo Mplab nativo di Microchip. Ciò comporta che, nel caso in cui si impieghi il compilatore MikroPascal (e stessa cosa vale per MikroC e MikroBasic) questi potranno essere utilizzati per caricare il file *.hex* nella memoria del controllore ma non per il debug. Per eseguire quest'ultima operazione sarà necessario disporre dell'ambiente Mplab di Microchip.

DEMO-BOARD E STARTER KIT CON E SENZA PROGRAMMATORE E DEBUGGER INTEGRATO

Particolarmente utili e flessibili per scopi

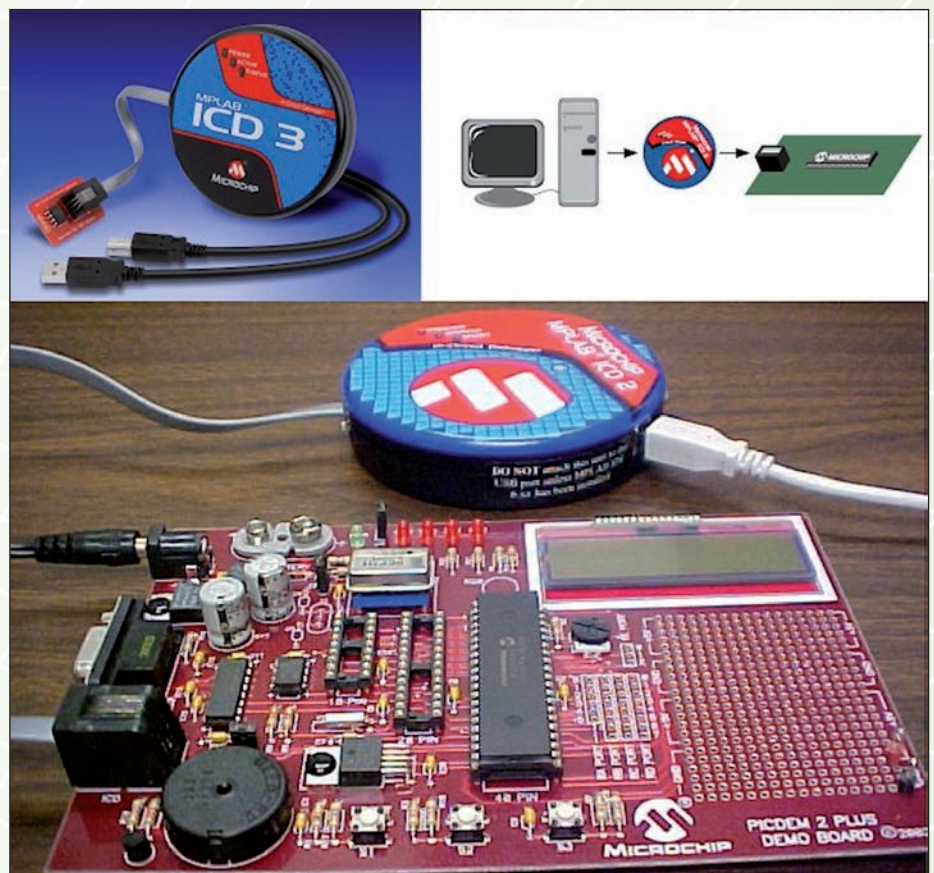



Figura 7-ICD2, ICD3, e connessione tra ICD e target-board che può coincidere, per esempio, con una demo-board come PICDEM 2 Plus [9].

Figura 8-PicKit2 e PikKit3 di Microchip [9]. Nella figura centrale è riportato il PicKit2 collegato a uno zoccolo di adattamento per la programmazione non in-circuit del controllore

didattici e spesso anche di sviluppo sono le demo-board e gli starter kit con programmatore ed eventualmente debugger integrato. Le *EasyPIC* di Mikroelektronika ne sono un esempio. L'integrazione on board di I/O digitali e analogici, di porte UART, SPI, I2C, USB, Ethernet e di visualizzatori a 7 segmenti, l'espandibilità con LCD alfanumerici, grafici e touch-panel e la predisposizione, tramite zoccoli, per il collegamento di PIC di diverse piedinature sia della serie 10F, 16F che 18F rendono queste schede estremamente flessibili anche grazie alla programmazione ed al debug in-circuit. Anche molti dei progetti di esempio mostrati fino a questo momento e di quelli che seguiranno sono stati testati con l'ausilio della scheda di sviluppo EasyPIC (precisamente con EasyPIC 3, una versione precedente alle versioni 5, 6 e 7 che come queste ultime supporta la programmazione in-circuit ma che differentemente da esse non integra il debugger). La serie di kit EasyPIC, giunta ormai alla versione 7 (Figura 9), è prodotta da Mikroelektronika e può inserirsi perfettamente sia in un percorso didattico che in un ambito progettuale di tipo professionale. Si tratta infatti a tutti gli effetti di kit di sviluppo che integrano sia il programmatore che numerose risorse hardware on-board utili per lo sviluppo e la prototipazione.

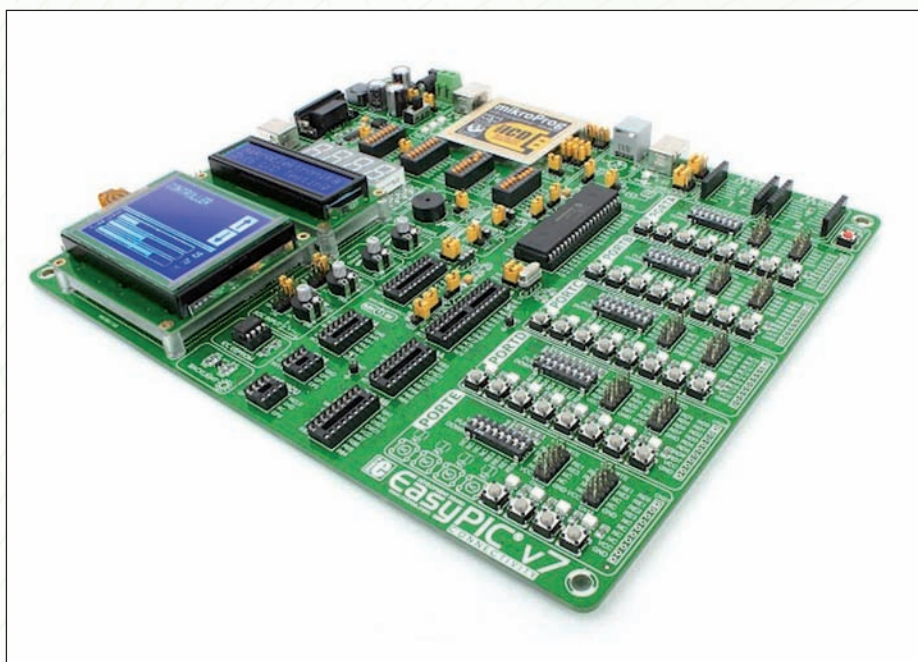



Figura 9-EasyPIC7 ultima della fortunata serie di schede EasyPIC di Mikroelektronika [10]

Come vedremo nei prossimi paragrafi tuttavia, in una fase iniziale del percorso didattico si può anche evitare di dotarsi di un kit di sviluppo colmando la lacuna utilizzando alcuni strumenti virtuali che consentono di simulare gli strumenti hardware.
Ciò non toglie che l'apprendimento degli argomenti trattati nelle lezioni del presente corso possa trarre notevole beneficio dalla disponibilità di kit di sviluppo come EasyPIC7. Si rimanda al sito del produttore [10] per maggiori approfondimenti. Anche altre demo-board meno assortite, per esempio non integranti il debugger oppure la programmazione in-circuit possono risultare estremamente utili allo scopo. E' il caso della diffusa *PICDem2 Plus* (Figura 7) o di altre simili demo-board di Microchip o di terze parti che nascono con la filosofia di essere integrate con un programmatore in-circuit-debugger esterno quali ICD2 o ICD3 (Figura 7) oppure PicKit2 e PicKit3 (Figura 8).

APPROFONDIMENTO SUGLI STRUMENTI DI DEBUG: GLI IN-CIRCUIT-EMULATOR (ICE)

Abbiamo già accennato al fatto che molti kit di sviluppo integrano la programmazione in-circuit e funzioni da debugger. Questo consente di superare le limitazioni, nell'analisi e test del firmware, intrinseche nell'utilizzo dei simulatori e debugger software integrati nell'ambiente di sviluppo e utilizzabili senza disporre di alcun hardware.
In pratica è sempre necessario, a un certo punto dello sviluppo, superare alcune limitazioni legate al debug puramente software in modo da spostarci a tutti gli effetti verso uno scenario più realistico che possa sfociare con successo con il vero e proprio test sul campo. Tali limitazioni sono legate soprattutto al fatto che il firmware, e quindi il controllore, interagisce spesso con altri dispositivi esterni in maniera non esattamente prevedibile, dal momento che gli eventi che determinano il funzionamento di questi ultimi sono spesso difficili da simulare in maniera appropriata e realistica. Da ciò deriva la necessità di disporre di debugger. Abbiamo già discus-

so degli In-Circuit-Debugger. Questi consentono l'esecuzione delle operazioni di debug sfruttando opportunamente, per le operazioni, alcuni pin (gli stessi utilizzati per la programmazione) e parte della memoria programma del controllore. Infatti, per eseguire le operazioni di debug l'in-circuit-debugger carica automaticamente, nella parte alta della memoria programma del controllore, un apposito programma necessario per gestire operazioni quali esecuzione step by step, monitoraggio delle variabili, inserimento di breakpoint, ecc...

Esistono sul mercato anche strumenti che prendono il nome di *In-Circuit-Emulator* (ICE) che utilizzano un hardware specializzato che in fase di sviluppo dell'applicazione HW/FW si sostituisce completamente al controllore su cui l'applicazione firmware è destinata a girare. *Mplab Real ICE* di Microchip ([9], Figura 10) per esempio è uno strumento di questo tipo. L'aspetto interessante di un *In-Circuit-Emulator* sta nel fatto che non toglie, ai fini del debug, alcuna risorsa al controllore. L'altro aspetto meno evidente ma altrettanto interessante sta nel fatto che mentre un in-circuit-debugger necessita sulla scheda di un connettore per il debug, che poi è lo stesso utilizzato per la programmazione in-circuit, l'emulatore normalmente non presenta questa limitazione dal momento che la sua interfaccia fisica con la scheda è costituita dall'intero pin-out dello stesso controllore che viene rimosso dalla board e interamente sostituito dall'emulatore stesso. Per questo motivo l'emulatore è normalmente connesso alla board attraverso un cavo che termina con un adattatore che cambia a seconda del controllore per il quale si sta sviluppando l'applicazione.

Questo consente di utilizzare lo stesso emulatore per qualunque controllore appartenente a determinate famiglie sostituendo il solo adattatore. Una piccola ma doverosa precisazione occorre fare per quanto concerne la compatibilità tra hardware e software degli strumenti di debug utilizzati. Per esempio, ICD2, ICD3 e Real ICE sono strumenti di debug di Microchip che per essere utilizzati necessitano dell'ambiente di sviluppo Mplab di Microchip e che non possono quindi essere utilizzati pienamente con altri ambienti di sviluppo. Essi possono inoltre essere utilizzati insieme alla maggior parte delle board di prototipazione di Microchip. Normalmente, gli in-circuit-emulator sono prodotti di fascia alta e quindi più costosi di strumenti quali ICD2 ed ICD3 che possono essere per questo considerati debugger dal costo più accessibile. In altre parole, il debugger in-circuit (ICD) è

Figura 10-In-Circuit-Emulator Mplab Real ICE di Microchip [9]

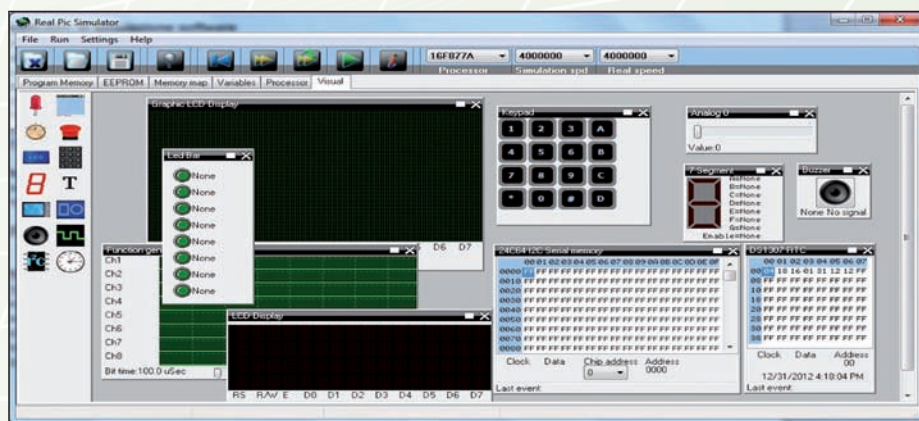



Figura 11-Interfaccia grafica di Real Pic Simulator ([11], [12])

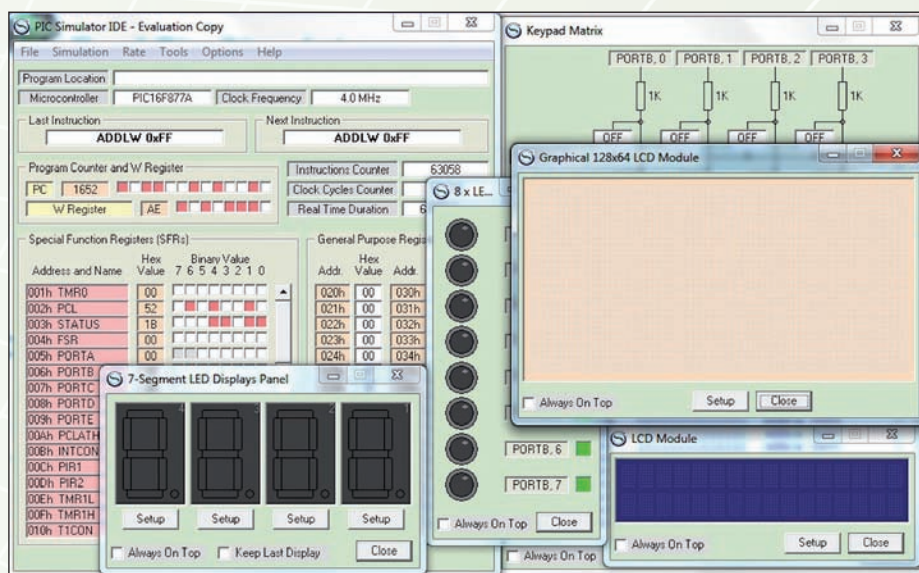



Figura 12-Interfaccia grafica di PIC Simulator IDE (Evaluation Version) [13]

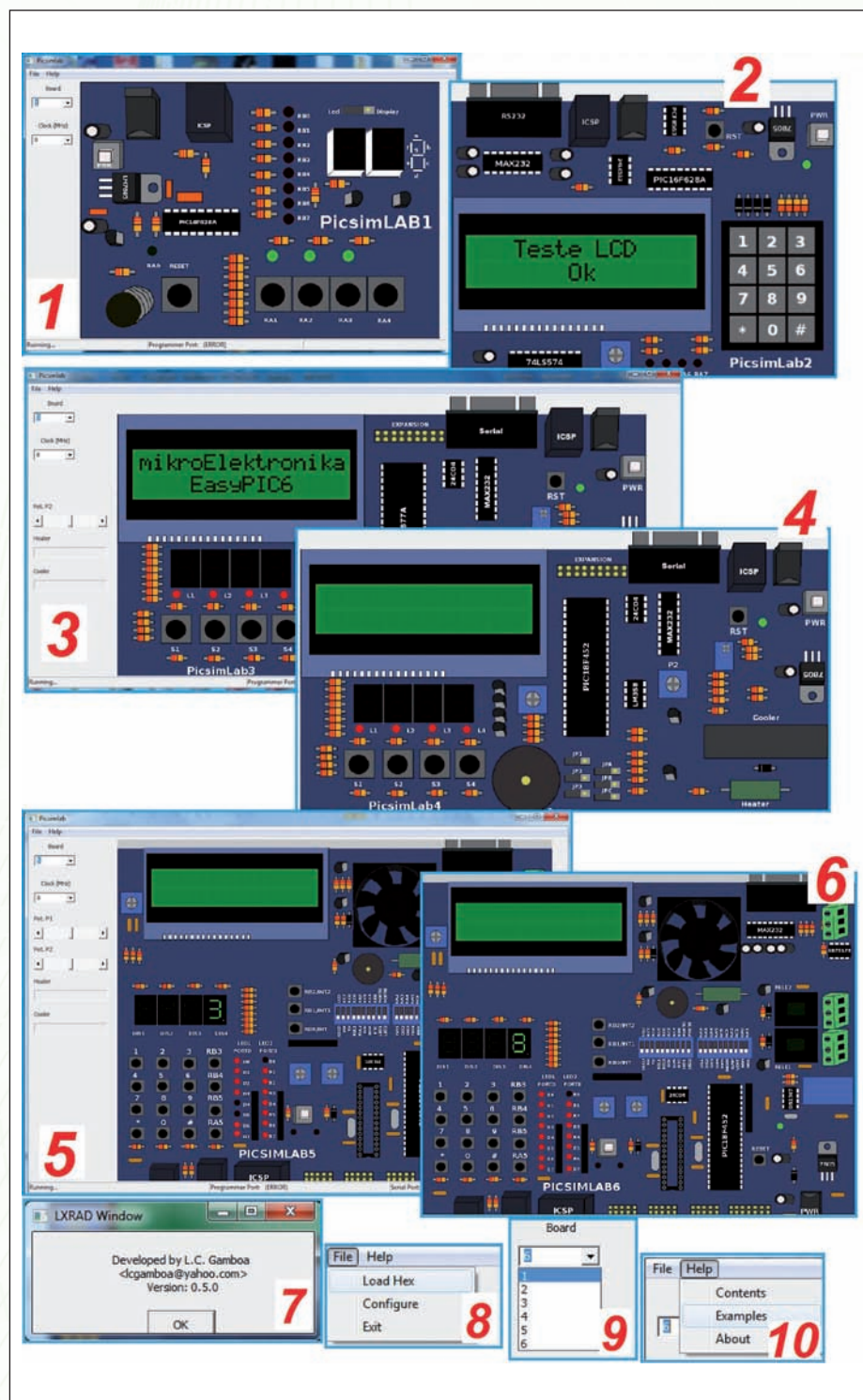



Figura 13-Le sei board virtuali integrate in PicSim LAB (1..6), i riferimenti dell'autore (7), il menu File da cui eseguire il caricamento virtuale del file eseguibile .hex (8), il menu a tendina per la selezione della board virtuale (9), l'help (10) da cui accedere ad esempi oppure agli schematici delle stesse board (Help/Contents).

un'alternativa economica all'emulazione in-circuit (ICE). Ovviamente, come già accennato, l'applicazione (scheda target su cui il controllore è montato) deve essere progettata con apposito connettore e connessioni al PIC per essere ICD compatibile e utilizzare controllori che integrano questo tipo di gestione. Il PIC 16F877 che abbiamo utilizzato ed utilizzeremo nella maggior parte degli esempi di progetto di questo corso, per esempio, è stato uno dei primi microcontrollori di Microchip ad implementare la tecnologi ICD. Per maggiori approfondimenti si rimanda all'analisi diretta dell'architettura hardware del PIC 16F877 ([3]; paragrafo 12.14-*In Circuit Debugger*, 14.8-*MPLAB ICD In-Circuit Debugger*) e ad articoli specifici quali [6], [7] e [8].

APPLICATIVI DI SIMULAZIONE SOFTWARE

Esistono applicativi software che consentono di simulare l'hardware del controllore o di un'intera demo-board virtuale eventualmente configurabile nei collegamenti (virtuali) che esistono tra i componenti. Strumenti di questo tipo consentono, soprattutto a livello didattico, di evitare la necessità di disporre di risorse hardware reali. Di seguito accenniamo a tre prodotti di questo tipo che potrebbero risultare particolarmente utili come ausilio al corso che stiamo sviluppando:

- *Real Pic Simulator (versione 1.1.0.0)* prodotto da DigitalElectroSoft ([11], [12]);
- *Pic Simulator IDE* (versione 6.40) prodotto da Oshonsoft (autore: Vladimir Soso) [13];
- *PicSimLab* prodotto free sviluppato dall'autore L.C. Gamboa [14].

REAL PIC SIMULATOR

Real PIC Simulator ([11], [12], Figura 11) è un applicativo che consente di configurare un hardware virtuale costruito intorno ad un controllore PIC delle famiglie 10F, 12F, 16F sul quale simulare le proprie applicazioni firmware. E' importante sottolineare che non si tratta di un software freeware, tuttavia è possibile utilizzarne gratuitamente una versione di prova per un periodo limitato di trenta giorni superati il quali è necessario acquistarne la licenza. Si tratta di uno strumento virtuale molto ben organizzato che consente di simulare Led, segnali analogici, display LCD alfanumerici 2x16, 2x20, 4x16 e 4x20, display GLCD Samsung KS107/KS108 compatibili, display a 7 segmenti, interfacce RS232, pulsanti, keypad 4x4, oscilloscopio virtuale, buzzer virtuale, PWM, generatore di funzioni digitali ad 8 canali, eeprom I2C 24C64, I2C Real Time Clock DS1307.
L'interfaccia (Figura 11) appare semplice e intuitiva e la configurabilità dei collegamenti tra pin del PIC e componenti hardware virtuali lo rendono indubbia-

mente molto flessibile. Tra i limiti vi è il fatto che non sono simulate alcune funzionalità, per esempio le istruzioni di scorrimento dei display LCD almeno nella revisione 1.1.0.0 provata per l'occasione in versione demo. Non si tratta tuttavia di un limite particolarmente importante e del resto va sempre considerato che la simulazione di una applicazione HW/SW non può mai essere esattamente equivalente in tutti gli aspetti al test dell'applicazione reale. Per maggiori dettagli e approfondimenti si rimanda al sito del produttore e all'impiego della versione di prova scaricabile dallo stesso sito [12], all'help in linea ed all'articolo [11].

PIC SIMULATOR IDE

E' un simulatore molto simile a Real PIC Simulator, altrettanto valido, che integra in linea di massima gli stessi strumenti hardware virtuali. In alcuni casi mostra una rapidità di esecuzione inferiore rispetto a Real PIC Simulator specie se si tiene a video l'interfaccia che consente di analizzare l'intera evoluzione di tutte le risorse fisiche interne del controllore (memoria, registri, ecc…). In compenso presenta la stessa configurabilità di Real PIC Simulator e consente la simulazione di alcune funzionalità non consentite in Real PIC simulator come le funzioni di shift nei display LCD. Per maggiori dettagli si rimanda al sito del produttore [13] e se ne consiglia l'impiego almeno della versione di prova scaricabile dal sito dello stesso produttore. Ovviamente, trattandosi di un prodotto coperto, come Real PIC Simulator, dai diritti d'autore, può essere utilizzato nella versione di prova solo per un periodo limitato dopo il quale è necessario l'acquisto della licenza d'uso.

PICSIMLAB

PICSimLab [14], sviluppato dall'autore L.C. Gamboa (Figura 13-7), è una applicazione software freeware che consente di simulare alcune demo-board per microcontrollori PIC di Microchip comparse nel recente passato sul mercato. La versione cui facciamo riferimento in questo paragrafo è la 0.5.0. Attraverso il menu a tendina *Board* (Figura 13-9) è possibile scegliere una tra 6 possibile demo-board virtuali. Ciascuna demo-board virtuale è caratterizzata da uno specifico modello di controllore virtualmente equipaggiato e da un certo numero di risorse hardware virtuali. Il firmware, una volta sviluppato e compilato, per esempio attraverso l'IDE MikroPascal Pro for PIC, viene virtualmente caricato sul controllore attraverso il comando *File/Load Hex* (Figura 13-8). A questo punto se ne può osservare l'effetto sulla board virtuale rappresentata a video graficamente.
E' possibile selezionare una delle seguenti demo-board virtuali:

- *PICSimLab1(Figura 13-1)*: è una board virtuale minimale per PIC16F628A che simula, oltre a detto controllore, le seguenti risorse hardware: interruttore power (ON/OFF), pulsante di reset, alcuni led, alcuni pulsanti e due display a 7 segmenti;
- *PICSimLab2(Figura 13-2)*: è una board virtuale minimale per PIC16F628A che consente di simulare le seguenti risorse hardware: interruttore power (ON/OFF), pulsante di reset, alcuni led, un tastierino a matrice 3x4, un display LCD alfanumerico 2x16, una porta RS232 virtualmente connessa al proprio PC, una eeprom esterna della serie 24CXX, un real time clock (RTC) PCF8563. E' interessante osservare il chip 74LS574 che consente di utilizzare due soli pin del PIC per il controllo del display. Per maggiori dettagli si rimanda allo schema elettrico della board accessibile attraverso *Help/Contents* (Figura 13-10);
- *PICSimLab3 (Figura 13-3)*: è una board virtuale per PIC16F877A che consente di simulare le seguenti risorse hardware: interruttore power (ON/OFF), pulsante di reset, alcuni led, quattro display a sette segmenti, altrettanti pulsanti, un display LCD alfanumerico 2x16, una porta RS232 virtualmente connessa al proprio PC, una eeprom esterna 24C04, un buzzer, un potenziometro analogico, una ventola di raffreddamento;
- *PICSimLab4 (Figura 13-4):* identica alla precedente ma con a bordo il PIC 18F452 al posto del 16F877A;
- *PICSimLab5 (Figura 13-5)*: è una board virtuale per PIC16F877A che consente di simulare le seguenti risorse hardware: power (ON/OFF), pulsante di reset, alcuni led, quattro display a sette segmenti, pulsanti, tastierino numerico, un display LCD alfanumerico 2x16, una porta RS232 virtualmente connessa al proprio PC, una eeprom esterna 24C04, due relè, due potenziometri analogici, una ventola di raffreddamento virtuale;
- *PICSimLab6 (Figura 13-6):* identica alla precedente ma con a bordo il PIC 18F452 al posto del 16F877A.

Attraverso *Help/Contents* è possibile accedere a una presentazione dell'appli-

cativo ed agli schemi elettrici di ciascuna board virtuale, mentre attraverso *Help/Example* è possibile accedere ai firmware .hex di esempio.
E' importante tenere a portata di mano gli schemi elettrici delle board virtuali al fine di stilare codice firmware che sia in accordo con essi.
Da questo punto di vista è infatti importante sottolineare che, le interconnessioni virtuali tra i componenti presenti su ciascuna board sono fissate da detti schemi elettici e possono essere configurate solo in minima parte, attraverso dip-switch virtuali, come accade tipicamente in una demo-board reale.

DUE UTILI OSSERVAZIONI

Da un punto di vista pratico occorre fare due importanti osservazioni sugli strumenti virtuali trattati negli ultimi tre paragrafi. La prima riguarda la configurabilità hardware (virtuale) che è certamente più flessibile in strumenti come Real PIC Simulatore e PIC Simulator IDE piuttosto che in strumenti come PICSimLab. I primi due infatti non sottintendono uno schema di collegamento predefinito tra i componenti ed il controllore. La board virtuale può cioè essere configurata, da un punto di vista hardware virtuale, a piacimento dall'utente che definisce, secondo le proprie necessità, le interconnessioni tra le risorse hardware virtuali disponibili ed il controllore. Questo conferisce allo strumento un'elevata flessibilità. Al contrario, PicSimLab presuppone, come riferimento, un hardware virtuale il cui schema elettrico è in buona parte fissato in partenza e che quindi prevede minori gradi di libertà nella relativa configurazione.
Una seconda osservazione va fatta a riguardo della completa o parziale rispondenza alla realtà della simulazione virtuale del firmware offerta da questi strumenti. Per esempio, se si prende in considerazione il controllo di un LCD, alcune istruzioni del set dell'LCD possono non essere contemplate. Il firmware di Esempio2 (Figura 3), per esempio, in simulazione funziona perfettamente con Real PIC Simulator mentre sembra non funzionare correttamente con PIC Simulator IDE. Analogamente l'Esempio3 (Figura 4) funziona nella simulazione virtuale perfettamente con Simulator IDE mentre sembra non funzionare correttamente con Real PIC Simulator.

TIRANDO LE SOMME...

Quanto esposto sugli strumenti di sviluppo, test, prototipazione, produzione, ci consente di tirare alcune semplici conclusioni che riassumiamo nei seguenti punti:

1. i programmatori per produzione appartengono a una linea di prodotti sicuramente inutile per chi intende condurre un percorso didattico di studio dei microcontrollori Microchip e di un ambiente di sviluppo come MikroPascal Pro for PIC;
2. stessa cosa dicasi per gli In-Circuit-Emulator (ICE) che sono strumenti di fascia alta e non entry-level e pertanto con performance e costi eccessivi per un percorso didattico;
3. gli In-Circuit Debugger, le demo-board, le evaluation-board ed i kit di sviluppo sono certamente da collocarsi come adeguate, a seconda dei casi, per scopi che possono spaziare dall'uso professionale a quello hobbistico e didattico essendo caratterizzati da flessibilità e costi molto più contenuti. In particolare, le schede EasyPIC sono strumenti che, ad un costo compreso tra 150 e 200 euro circa, mettono a disposizione tutto il necessario per scopi didattici e di sviluppo, sia in termini di risorse hardware che di funzionalità di programmazione e debug in-circuit;
4. i simulatori software come *Real Pic Simulator, PICSimulator IDE e PicSimLab* possono consentire un primo approccio semplice ed efficace con la materia della programmazione dei microcontrollori PIC di Microchip a prescindere dal particolare ambiente di sviluppo che si sta utilizzando, a costi ancora più contenuti, in alcuni casi praticamente nulli se si impiegano, almeno inizialmente e per un periodo limitato, le relative versioni demo.

Il consiglio che può ritenersi più appropriato, sia in ambito didattico che in ambito semi-professionale, è quello di rivolgere l'attenzione verso demo-board e kit di sviluppo con in-circuit-debugger di costo relativamente contenuto come appunto EasyPIC senza disdegnare strumenti di simulazione software come quelli richiamati al precedente punto quattro (almeno per quanto riguarda l'approccio didattico). Man mano che l'acquisizione dei concetti e la relativa applicazione pratica degli stessi procede, passando eventualmente da un approccio didattico a uno semiprofessionale o professionale, l'attenzione dovrebbe rivolgersi sugli strumenti riportati nei precedenti quattro punti in ordine inverso rispetto a come sono stati elencati.

CONCLUSIONI

In questa puntata abbiamo concluso l'argomento del controllo dei display LCD alfanumerici attraverso controllori PIC programmati in MikroPascal Pro for PIC parlando di *ASCII Chart* e del *LCD Custom Character*. Quanto esposto è stato corredato, come al solito, di alcuni semplici ma mirati codici MikroPascal di esempio. Nella seconda parte della lezione abbiamo trattato l'argomento degli strumenti di sviluppo HW/SW per controllori PIC che possono essere utili, oltre all'IDE MikroPascal, per eseguire esperienze pratiche e rendere più divertente ed efficace l'acquisizione sia dei concetti trattati fino a questo momento che di quelli che saranno trattati nelle prossime lezioni. A tale proposito si è cercato di mettere in evidenza le principali caratteristiche di questi strumenti al fine di comprendere quali siano più completi e quali meno, quali più vicini ad un ambito didattico e quali ad un ambito professionale, quali più costosi e quali più accessibili.
Sebbene, come abbiamo avuto modo di comprendere, lo studio dei controllori PIC e di MikoPascal Pro for PIC non comporti necessariamente l'acquisto di una demo-board per fare esperienze pratiche, soprattutto entry-level, si preferisce spesso acquistarne una al fine di prendere dimestichezza, anche fisicamente, con l'hardware e rendere più tangibile la messa in pratica di quanto appreso e di quanto sviluppato in firmware.
Nella prossima puntata allargheremo ulteriormente l'orizzonte delle applicazioni pratiche mostrando come pilotare con il PIC, mediante codice MikroPascal Pro for PIC, display a sette segmenti e come realizzare l'hardware ed il firmware di applicazioni quali contatori, contapezzi e contagiri. ❏

CODICE MIP 2835331

di DANIELE CAPPA

IL SUPER VXO

Potrebbe essere l'oscillatore locale di un ricevitore minimo, o di un piccolo RTX QRP. Questi sono gli appunti circa l'utilizzo di un VXO modificato.

SVXO acronimo in Inglese di Variable Xtal Oscillator, ovvero oscillatore variabile controllato a cristallo, sui tratta quindi di una contraddizione di fatto.. un oscillatore variabile controllato a quarzo.

Fino alla metà degli anni '70 l'oscillatore a cristallo era l'unica soluzione circuitale economica per ottenere un oscillatore con eccellenti performance in fatto di stabilità. Poi PLL e sintetizzatori hanno cambiato le carte in tavola, ma per l'hobbysta è cambiato poco. Sistemi più moderni, e performanti fanno spesso a pugni con la semplicità circuitale e per piccoli ricevitori, o ricetrasmettitori, si ricorre sempre al classico VFO, o al meno noto PTO di cui si è già parlato su queste pagine (rivista n. 274).

Il VXO rappresenta una delle possibili alternative, lo spostamento della frequenza di oscillazione del quarzo dipende da una bobina in serie al quarzo in parte compensata da un condensatore variabile, anche lui in serie al quarzo. In queste condizioni lo spostamento è di solito modesto, alcuni KHz, non di più.

Verso la metà degli anni '70 questo sistema era utilizzato per la sintonia di apparati amatoriali portatili. Tra questi sicuramente il più noto è IC202 di casa ICOM che utilizzava un VXO per spostare la sintonia della radio in un range di 200 KHz, la sottogamma 2 metri SSB era dunque coperta con 5 quarzi (se non sbaglio ne montava al massimo 4…). L'escursione di 200 KHz era ottenuta spostando di circa 23 KHz un oscillatore la cui uscita veniva successivamente moltiplicata per 9, ottenendo quindi i citati 200 KHz. Questo sistema è difficilmente proponibile in un progetto casalingo, oltretutto si ottengono frequenze decisamente più alte delle nostre necessità legate all'oscillatore locale di un piccolo RX.o RTX in HF.- La soluzione che è ormai citata dovunque come "super VXO" sfrutta quella proposta a metà del 1980 da due radioamatori giapponesi, JA0AS, ora SK e JH1FCZ, hanno avuto una idea brillante quanto facile da realizzare. Hanno utilizzato alcuni quarzi uguali posti in parallelo tra loro. Con questo sistema lo spostamento aumenta molto, esemplari da 14 Mhz in fondamentale è in grado di spostarsi di 400 e più KHz.

Foto 1: i quarzi

CI PROVO…

Anche in questo caso la realizzazione è portata a termine sfruttando quanto offre il cassettino dei recuperi, dunque i quarzi sono da 14,318 MHz, provenienti da

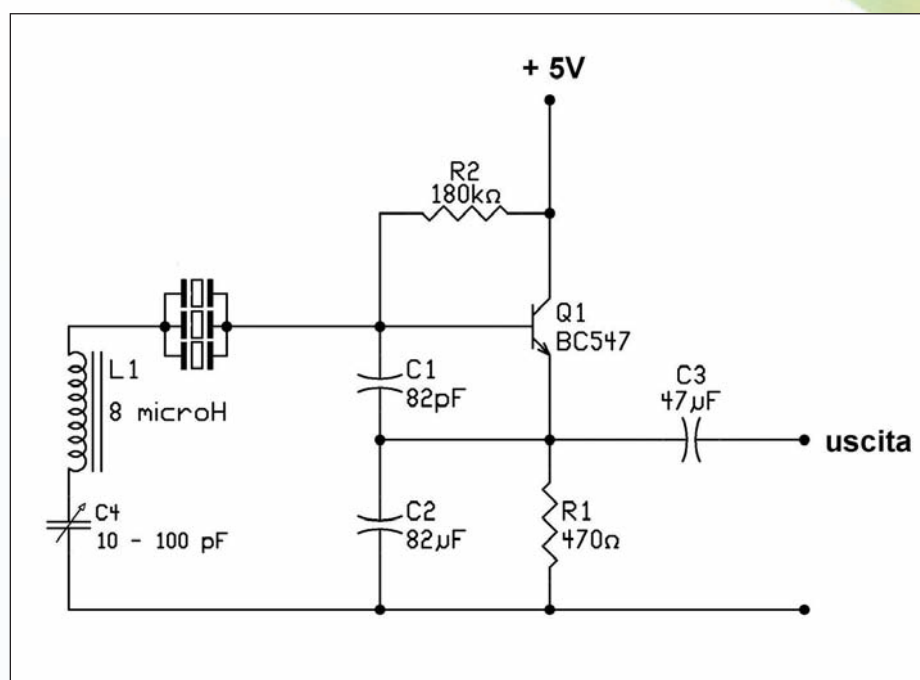



Figura 1 – Lo schema elettrico

Foto 2 – La prova

vecchie VGA, la bobina è recuperata da un alimentatore a commutazione da PC, altri esemplari sono realizzati con il filo da 0,7 mm recuperato da un vecchio trasformatore e avvolto su due strati su un diametro di 7,5 mm, utilizzando una matita quale supporto provvisorio. Il doppio strato è realizzato avvolgendo metà spire direttamente sulla matita, quindi le fermiamo con qualche centimetro di nastro di carta, torniamo indietro con il filo dell'avvolgimento e iniziamo il secondo strato. I due strati sono avvolti entrambi nello stesso senso e iniziano entrambi dallo stesso lato.

Lo strato esterno andrà fermato con qualche goccia di colla, cianoacrilica o bostik, basta stia fermo… La bobina è successivamente sfilata dalla matita-supporto. Il risultato sarà una bobina piuttosto massiccia, lunga un paio di centimetri, libereremo dallo smalto i primi centimetri del filo di rame utilizzato il solito coltellino.

Sono stati utilizzati tre quarzi uguali, posti in parallelo tra loro, il condensatore variabile, per le prove non ho evidentemente utilizzato un varicap, è un esemplare demoltiplicato da 400 pF, anche se la variazione avviene entro i primi 100 pF.

In queste condizioni con una bobina da 4 microH abbiamo una variazione modesta, 12 Khz circa, che salgono a oltre 400 KHz se la bobina è da 8 microH… il problema è che più ci si allontana dalla frequenza tipica del quarzo e meno l'oscillatore è stabile.

Dunque a 50 Khz la stabilità è eccellente, a 100 Khz è buona, ma non eccezionale… andando oltre si avvertono i problemi di un normale VFO dotato di oscillatore libero.

Una cosa importante è che la variazione avviene quasi completamente verso il basso, dunque il citato parallelo di tre quarzi da 14,318 MHz "scende" fino sotto i 14 MHz, ma sale di pochissimo, praticamente nulla.

La situazione contraria si verifica se la bobina non ha una impedenza sufficiente, dunque con una bobina troppo piccola, o senza bobina, la variazione di frequenza avviene verso l'alto, ma questa è limitata a 12 – 15 Khz.

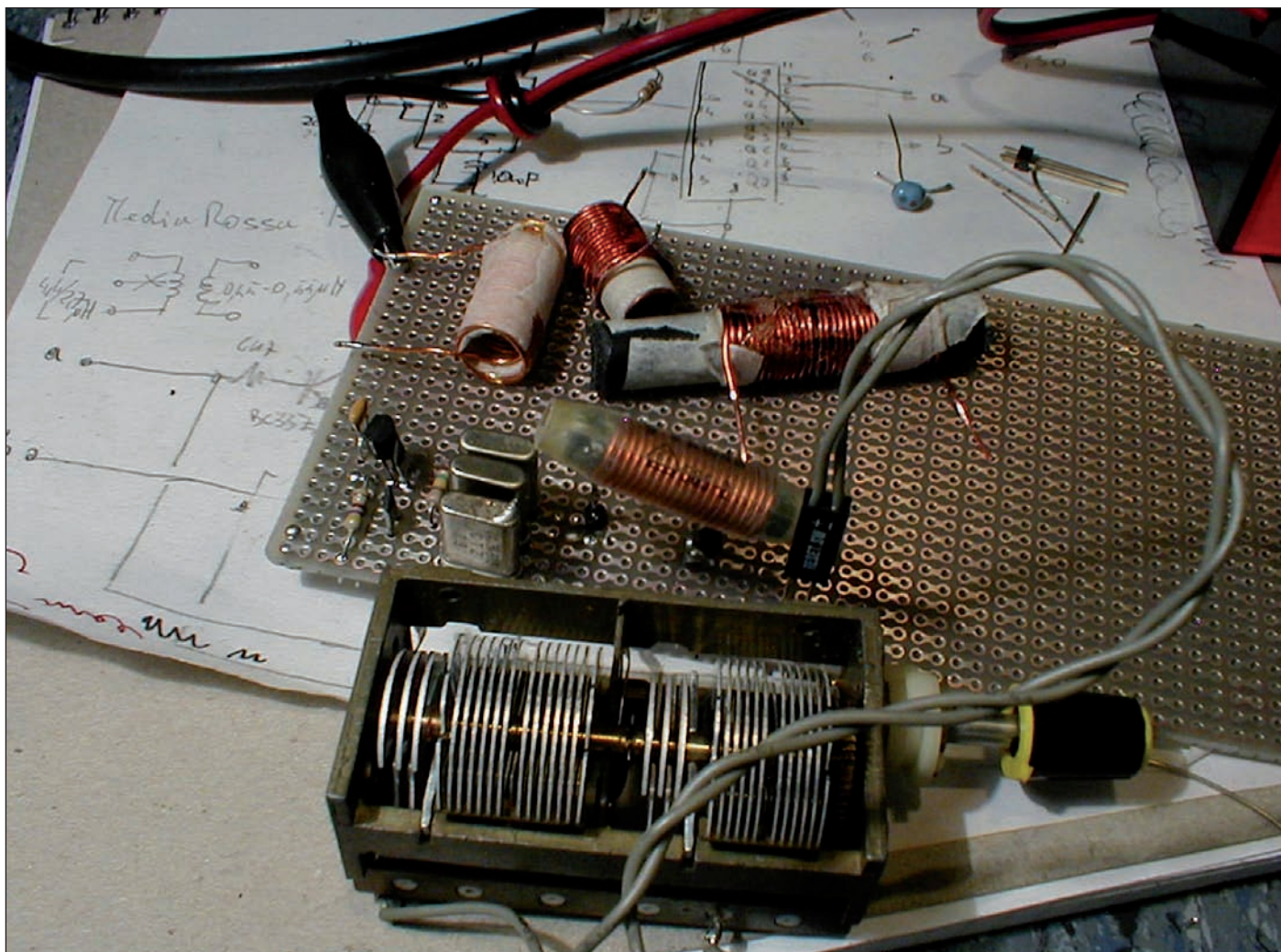



Foto 3: Il circuito di test

Queste variazioni dipendono ovviamente dalle caratteristiche dei quarzi impiegati, dunque sono soggette a differenze anche molto ampie in base ai quarzi impiegati.
Le prove sono state portate a termine impiegando alcune bobine avvolte in aria con valori di induttanza crescente, 4, 6 e 8 microH. Le bobine sono state calcolate con il solito radioutilitario e successivamente “aggiustate” con l’induttanzimetro.
L’oscillatore è costruito su un banalissimo BC547, con due cond. NP0 da 82 pF come reazione tra emettitore e base, questi condensatori non sono critici, qualasi valore tra 47 a 100 pF va ugualmente bene . L’uscita è prelevata in emettitore, per prelevare il segnale di uscita qui andrebbe posto un ceramico di basso valore, per non complicare la vita dell’oscillatore. In realtà i test sono stai condotti semplicemente ascoltando la nota di battimento sul ricevitore di stazione. Per l’utilizzo definitivo sarà indispensabile provvedere a un ulteriore stadio quale buffer di uscita. Lo schema elettrico visibile nella figura è assolutamente di prova, il condensatore variabile dovrà essere preceduto da un buon trimmer ceramico che ne limita l’escursione.
L’alimentazione andrà ripulita a dovere, dunque una impedenza in collettore , oppure una resistenza di basso valore (22 – 100 ohm) preceduta da un ceramico da 47 – 100 nF verso massa.
L’uso definito dovrebbe inevitabilmente prevedere un diodo varicap in luogo del vecchio, ingombrante e introvabile variabile ad aria… Un potenziometro multigiri e il gioco è fatto! Anche se nulla vieta di impiegare un variabilino a polimeri, quelli montati sulle radioline_spazzatura cinesi.
Anche l’impiego di un fet come oscillatore potrebbe essere una buona idea… insomma, si tratta di un’idea, una prova in vista di possibili sviluppi futuri, senza alcuna utilità pratica attuale…
Una curiosità… in gruppo quarzi – bobina – condensatore, così come disegnato funziona su altri oscillatori… su un vecchio SA602, mixer bilanciato di buona memoria, funziona perfettamente, e in modo assolutamente analogo a quanto rilevato sull’oscillatore a transistor. L’idea termina qui, queste righe vogliono essere il resoconto di una prova volta verificare quanto si legge in rete circa questa tipologia di VXO. Il problema nell’utilizzo pratico rimane sempre il medesimo… è necessario reperire alcuni quarzi adatti, e di solito i valori comunemente reperibili tra le categorie economiche sono ben lontani dalla frequenza che ci serve. ❏

CODICE MIP 2835900

THE PREMIER BUSINESS DEVELOPMENT EVENT FOR THE GLOBAL ROBOTICS INDUSTRY

GENOA 2013 | APRIL 11-12

MAGAZZINI DEL COTONE - PORTO ANTICO | GENOA ITALY

WWW.ROBOBUSINESS.EU

ORGANISED BY:

RBE is the European edition of:
ROBOBUSINESS LEADERSHIP SUMMIT

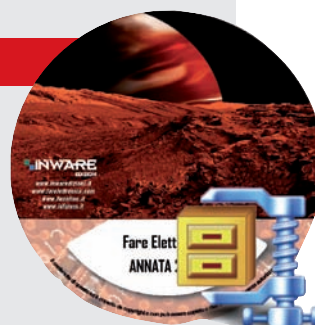
Lo shop dei prodotti Inware Edizioni...

Nuova piattaforma!

PRIMO PIANO

ANNATA Fare Elettronica 2012

In un comodo file zip da scaricare, tutti i numeri di Fare Elettronica del 2012 in pdf ad alta risoluzione! Potrai così visualizzare le singole riviste e stampare gli articoli che più ti interessano senza perdita di qualità. Nel file zip è incluso anche una comodissima interfaccia utente che ti permetterà di navigare nelle riviste con estrema facilità.

CODICE: FE-CD2012 PREZZO: € 9,99

NOVITÀ

Display 1.5" microOLED

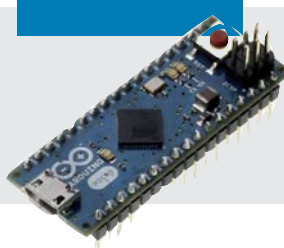
Il μOLED-128-G2 è un modulo display intelligente che serve come la soluzione perfetta per display embedded per qualsiasi progetto che richiede la brillantezza del colore, animazione o immagini tutto su uno schermo di 1.5" OLED.
Il cuore di questo modulo è il processore grafico GOLDELOX, che è guidato da un virtual core engine altamente ottimizzato: EVE (Extensible Virtual Engine). Una vasta gamma di periferiche hardware e software sono stati integrati in questo modulo, in maniera tale da dare all'utente la libertà di adattare il modulo per ogni qualsiasi applicazione, troviamo, ad esempio, un display PMOLED da 1.5", un connettore per card micro-SD, GPIO che possono essere utilizzati come input/ouput digitali, ingresso analogico, One-Wire e generazione del suono.

CODICE: UOLED-128-G2 PREZZO: € 53,24

Arduino Micro

NOVITÀ

Arduino Micro è una scheda basata sul processore ATmega32u4, sviluppata in collaborazione con Adafruit. Possiede 20 pin input/output digitali (dei quali 7 possono essere usati come PWM output e 12 come input analogici), 16MHz di velocità di clock, un connettore micro USB, un connettore ICSP e un tasto per il reset. Contiene tutto il necessario per supportare il microcontrollore; per utilizzarlo, connettetelo a un PC via USB o alimentatelo via adattatore AC>DC o tramite batterie. La sua forma è stata studiata apposta per essere facilmente inserita nella breadboard.

CODICE: A000053 PREZZO: € 21,78

NOVITÀ

EasyPIC7 for dsPIC30

La EasyPICv7 for dsPIC30 è la settima generazione della famosa scheda di sviluppo per dsPIC.
E' provvista di moduli di cui avrete bisogno per lo sviluppo delle vostre applicazioni DSP: programmer e debugger on-board, supporto CAN, sensori di temperatura, connettori per display e sei gruppi di I/O sono solo alcuni dei moduli presenti su questa scheda. La scheda viene fornita con un dsPIC30F4013.

CODICE: EASYPIC7-DSPIC30 PREZZO: € 175,45

GPS Click

NOVITÀ

GPS Click è una scheda accessoria che utilizza il nuovo socket mikroBUS ed è una soluzione compatta per aggiungere le funzionalità GPS sul vostro dispositivo. E' dotata di un motore di posizionamento, LEA-6S, u-blox 6 ad alte prestazioni. La scheda può essere interfacciata con un microcontrollore tramite connessione UART o I2C, oppure i dati possono essere acquisiti utilizzando una applicazione per PC attraverso la connessione USB. Una delle caratteristiche della scheda è che il connettore è compatibile con le antenne attive e passive. La GPS Click può utilizzare come tensione di alimentazione solo 3,3 V.

CODICE: MIKROE-1032 PREZZO: € 50,82

OFFERTA

Smart Lcd Display Module

L'uLCD-28PT (SGC) è un modulo display 2,8" seriale all-in-one economico e compatto che fa uso dell'ultimo ritrovato in fatto di tecnologia LCD (TFT) con un controller grafico seriale PICASO-SGC integrato che fornisce funzionalità stand-alone a qualsiasi progetto. Grafica molto potente, testi, immagini, animazioni e altre innumerevoli caratteristiche sono presenti all'interno del modulo. Offre un'interfaccia seriale semplice ma efficace a qualsiasi host microcontrollore che può comunicare tramite porta seriale. Tutte le funzioni relative allo schermo sono comunicate tramite un semplice protocollo via interfaccia seriale.

CODICE: ULCD-28-PT PREZZO: ~~€ 70,18~~ € 59,29

OFFERTA

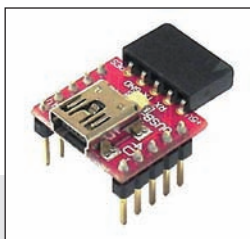
Micro-Oled 96X64

Il microOLED µOLED-96 è un display grafico a colori completo di scheda di controllo che può essere gestito direttamente da un microcontrollore (PIC, AVR, ARM, Stamp) via porta seriale. Possibilità di scegliere 3 tipi di dimensioni per il font (5x7, 8x8 and 8x12) e con appositi comandi disegnare cerchi, linee, rettangoli, scegliere il colore dello sfondo e caricare immagini bitmap. A bordo adattore memory card per micro-SD (µSD) per immagazzinare icone, immagini, animazioni ecc..

CODICE: UOLED-96-G1 PREZZO: ~~€ 47,10~~ € 35,09

Convertitore USB/RS232

Il microUSB (USB-MB5) è un convertitore bridge da USB a RS232 molto semplice, piccolo economico e facile da usare. Questo dispositivo utilizza un bridge CP2102 della Silicon LAB e un connettore mini-USB per il collegamento con un PC. Il prodotto fornisce il multi Baud Rate seriale, il controllo di flusso in un package Dual-In-Line a 10 pin 0,1". Il µUSB è ideale per prototipi o per la produzione.

CODICE: UUSB-MB5 PREZZO: ~~€ 24,20~~ € 20,57

OFFERTA

OFFERTA

Sensore di GAS

Il Modulo Sensore Gas C2H5OH (Alcohol / Benzine) è disegnato per permettere a un microcontrollore di determinare il momento in cui il livello di gas C2H5OH precedentemente settato è stato raggiunto o superato. Il collegamento al modulo sensore è operato grazie a un SIP header 4-pin e richiede 2 pins I/O dal microcontrollore host. Il modulo sensore è principalmente indirizzato a fornire un mezzo di comparazione delle fonti di alcohol e ad essere in grado di settare un allarme qualora la fonte eccede.

CODICE: 27904 PREZZO: ~~€ 34,78~~ € 22,99

MAKER Card

Acquistando la MAKER Membership card avrai diritto a:

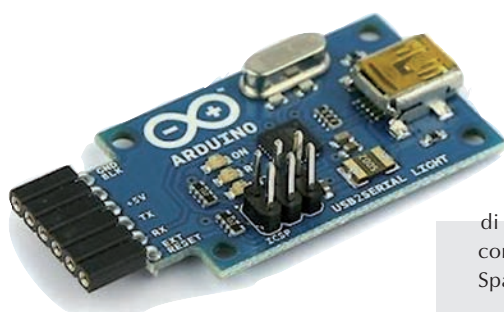
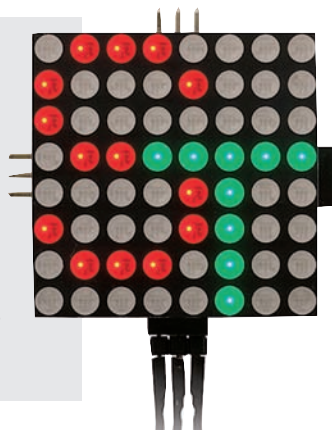
1) 11 nuovi numeri di Fare Elettronica sia in edizione cartacea che in PDF (in coda all'eventuale abbonamento esistente)
2) Accesso al club della rivista Fare Elettronica (centinaia di articoli in PDF)
3) Un e-book a scelta (Smart card, CPLD, display, PIC, AmpOP, alimentatori, linguaggio C, linguaggio BASIC, Pillole di elettronica Analogica)
4) Una raccolta delle annate di Fare Elettronica a scelta (dal 2003 al 2012)

CODICE: MIKROPROG PREZZO: € 49.50

Lo shop dei prodotti Inware Edizioni... ma non solo!

Modulo Display LED a matrice RGB 8x8

Aggiungi 64 LED colorati e brillanti nel tuo prossimo progetto! Ogni pixel nel display a matrice 8x8 può visualizzare milioni di colori diversi. Basato sul Propeller P8X32A, il display è pre-programmato con un firmware ricco di funzionalità e supportato da un set di comandi dettagliati. E' possibile controllare il modulo dai pin di I/O di un singolo un microcontrollore.

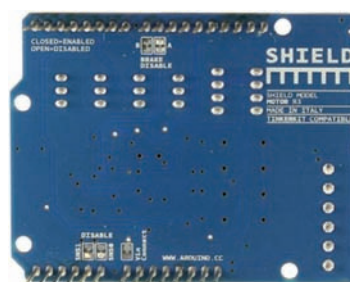
CODICE: 32324 PREZZO: € 38,72

Convertitore USB 2 Serial per Arduino

Questo modulo trasforma la porta USB in una porta COM. E' compatibile con il cavo FTDI a 5V. Permette di alimentare e programmare una Arduino Ethernet da qualunque scheda Arduino tramite il connettore di programmazione compatibile ai cavi FTDI. E' equipaggiato con lo stesso processore di Arduino Uno, che, attraverso all'interfaccia ICSP, permette di essere riprogrammato quando messo in DFU mode. Il connettore è compatibile con le prese standard offerte da connettori FTDI, Adafruit e Sparkfun.

CODICE: A000059 PREZZO: € 12,10

microLCD 2.4" con Touch Resistivo

L'LCD-24PTU è un modulo Display Intelligente compatto ed economico dotato di tantissime funzioni, pronto a diventare interfaccia grafica per alcune applicazione target e in grado di essere una interfaccia di controllo per altre applicazioni. Il cuore di questo modulo è il processore PICASO, che è guidato da un virtual core engine altamente ottimizzato: EVE (Extensible Virtual Engine). Una vasta gamma di periferiche hardware e software sono stati integrati in questo modulo, in maniera tale da dare all'utente la libertà di adattare il modulo per ogni qualsiasi applicazione.

CODICE: ULCD-24-PTU PREZZO: € 59,29

Arduino Motor Shield Rev3

L'Arduino Motor Shield si basa sull'integrato L298, che è un dual full-bridge driver progettato per pilotare carichi induttivi come relè, solenoidi, motori passo-passo e DC. Abbinato ad una scheda di sviluppo Arduino consente di pilotare due motori DC, controllando la velocità e la direzione di ciascuno in modo indipendente. Tra le altre caratteristiche, è inoltre possibile misurare l'assorbimento di corrente di ciascun motore. Lo shield dispone anche di connettori TinkerKit, il che significa che è possibile creare rapidamente progetti inserendo moduli TinkerKit sugli appositi connettori.

CODICE: A000079 PREZZO: € 24,20

prossimamente su Fare Elettronica

COMING...

ALIMENTATORE SWITCHING DA LABORATORIO

Alimentare uno strumento digitale professionale o caricarne le sue batterie dedicate richiede una priorità importante: l'affidabilità in condizioni di sovraccarico e cortocircuito. Vediamo nel dettaglio come e cosa fare per una buona qualità della tensione d'uscita e in che modo realizzare/dimensionare l'induttore principale.

PNEUMATICA CON CUBLOC

Un esempio di applicazione alla pneumatica dei microcontrollori con l'impiego del PC per l'elaborazione e la memorizzazione dei dati.

CONTROLLO MOTORE CON PONTE AD H

Ecco come possono essere controllati i motori in corrente continua di una certa potenza, con la possibilità di inversione del verso di rotazione, con la tecnica di controllo a ponte H.

CONTI CORRENTI POSTALI - Ricevuta di Versamento   BancoPosta

€ sul C/C n. 70107552   di Euro

importo in lettere

INTESTATO A: INWARE EDIZIONI SRL

ESEGUITO DA:
VIA - PIAZZA
CAP
LOCALITÀ

BOLLO DELL'UFF. POSTALE

AVVERTENZE
Il Bollettino deve essere compilato in ogni sua parte (con inchiostro nero o blu) e non deve recare abrasioni, correzioni o cancellature.
La causale è obbligatoria per i versamenti a favore delle Pubbliche Amministrazioni.
Le informazioni richieste vanno riportate in modo identico in ciascuna delle parti di cui si compone il bollettino.

CAUSALE

CONTI CORRENTI POSTALI - Ricevuta di Accredito   BancoPosta

€ sul C/C n. 70107552   di Euro

TD 451

importo in lettere

INTESTATO A: INWARE EDIZIONI SRL

CAUSALE

ESEGUITO DA:
VIA - PIAZZA
CAP
LOCALITÀ

AUT. DB/SISB/E 20017 DEL 11.04.2006

BOLLO DELL'UFF. POSTALE
codice bancoposta

IMPORTANTE: NON SCRIVERE NELLA ZONA SOTTOSTANTE
importo in euro   numero conto   td

70107552 < 451>

Indagine sui lettori

Aiutaci a conoscerti meglio!

Con il tuo aiuto riusciremo ad offrirti una rivista sempre più in linea con le tue aspettative.

Compila online il questionario all'indirizzo www.farelettronica.com/survey

Per ringraziarti per il tuo tempo e la tua cortesia, ti invieremo gratuitamente un bellissimo eBook del valore di 14,52 euro!

IMPORTANTE: NON SCRIVERE NELLA ZONA SOPRASTANTE!

AVVERTENZE

Il Bollettino deve essere compilato in ogni sua parte (con inchiostro nero o blu) e non deve recare abrasioni, correzioni o cancellature.

La causale è obbligatoria per i versamenti a favore delle Pubbliche Amministrazioni.

Le informazioni richieste vanno riportate in modo identico in ciascuna delle parti di cui si compone il bollettino.

PRODOTTI

Se vuoi ordinare "offline" ovvero senza usare Internet, usa il seguente modulo d'ordine **PRODOTTI**

COD. PRODOTTO	DESCRIZIONE	PREZZO UNITARIO	Q.tà	Totale
			SUB TOTALE	
			SPESE DI SPEDIZIONE	
			TOTALE	

Nota. Se le righe non sono sufficienti, utilizzare più copie del modulo.
Per maggiori dettagli sulle spese di spedizione e i metodi di pagamento vedi retro cartolina.

METODI DI PAGAMENTO

☐ **BONIFICO BANCARIO** ☐ **BOLLETTINO POSTALE** ☐ **CONTRASSEGNO**

☐ MasterCard ☐ VISA ☐ postepay

TITOLARE. . . . .
NUMERO . . . .
DATA DI SCADENZA. . . . . CODICE DI SICUREZZA . . . .

FE 332

INVIARE IL MODULO D'ORDINE A: Inware Edizioni srl, Via Giotto, 7 - 20032 Cormano (MI) - oppure via FAX allo 02 66508225
(Inviando questo modulo si accettano i termini e le condizioni riportate a tergo)

Nome . . . . Cognome . . . .
Indirizzo . . . . n° . . . .
Cap . . . . Città . . . . Prov . . . .
Tel . . . . Fax . . . . Email . . . .
Ragione Sociale . . . .
Codice fiscale . . . . Partita IVA . . . .

DATA . . . . FIRMA . . . . ☐ Fattura

ABBONAMENTO

Sì, voglio diventare membro o rinnovare la membership

☐ MAKER MEMBERSHIP: 11 nuovi numeri di Fare Elettronica + Bonus pack "MAKER" (vedi retro) € 49,50
☐ SMART MEMBERSHIP: 11 nuovi numeri di Firmware + Bonus pack "SMART" (vedi retro) € 39,50
☐ GENIUS MEMBERSHIP: 11 nuovi numeri sia di Firmware che di Fare Elettronica + Bonus pack "GENIUS" (vedi retro) € 79,50

FE 332

METODI DI PAGAMENTO (per maggiori dettagli vedi retro cartolina)

☐ **BONIFICO BANCARIO** ☐ **BOLLETTINO POSTALE** ☐ **ALLEGO ASSEGNO** (Intestato a Inware Edizioni)

☐ MasterCard ☐ VISA ☐ postepay

PayPal fare il pagamento su paypal@inwaredizioni.it

TITOLARE. . . . .
NUMERO . . . .
DATA DI SCADENZA. . . . . CODICE DI SICUREZZA . . . .

INVIARE IL MODULO D'ORDINE A: Inware Edizioni srl, Via Giotto, 7 - 20032 Cormano (MI) - oppure via FAX allo 02 66508225
(Inviando questo modulo si accettano i termini e le condizioni riportate a tergo)

Nome . . . . Cognome . . . .
Indirizzo . . . . n° . . . .
Cap . . . . Città . . . . Prov . . . .
Tel . . . . Fax . . . . Email . . . .
Ragione Sociale . . . .
Codice fiscale . . . . Partita IVA . . . .

DATA . . . . FIRMA . . . . ☐ Fattura

ISTRUZIONI PER L'ORDINE E SPESE DI SPEDIZIONE

Il modulo d'ordine dovrà essere compilato in tutte le sue parti ed inviato via posta o via fax ai recapiti indicati sul modulo d'ordine stesso. Gli ordini potranno essere fatti anche direttamente online dal sito www.elettroshop.com/ieshop. In questo caso non sarà necessario inviare il modulo d'ordine. Le spese di spedizione ammontano a euro 8,50 a cui vanno aggiunti euro 3,50 se si sceglie di pagare in contrassegno.

METODI DI PAGAMENTO

Si accettano pagamenti ino in contrassegno, carta di credit, bollettino postale o bonifico all'ordine. Per il contrassegno verrà applicata una spesa aggiuntiva di euro 3,50 per le spese di contrassegno. Forme diverse di pagamento devono essere previamente concordate.

Bonifico bancario

Appoggiato su Poste Italiane IBAN: IT 68 I 07601 01600 000070107552 intestato a Inware Edizioni srl.

Bollettino postale

Versamento sul conto corrente postale n. 70107552 intestato a Inware Edizioni srl. Inviare la ricevuta (o copia) del versamento.

Contrassegno

La merce verrà pagata direttamente al corriere alla consegna della merce. Il pagamento in contrassegno comporta l'addebito di euro 3,50 per spese di contrassegno.

Carta di Credito

Il pagamento con carta di credito può essere effettuato specificando i dati della carta via posta, email, web, fax o telefono. Per gli ordini effettuati sul sito il pagamento mediante carta di credito verrà effettuato attraverso una connessione sicura SSL per garantire la massima sicurezza.

TERMINI E CONDIZIONI

Tutti i prodotti sono coperti da garanzia. La garanzia copre tutti i difetti di fabbricazione entro un anno dal ricevimento della merce. Tutti i prodotti non funzionanti per uso improprio o incuria non saranno ritenuti in garanzia e saranno addebitati gli eventuali costi di riparazione. Tutti i prodotti verranno riparati e/o sostituiti direttamente dal produttore. Non sono coperti da garanzia i componenti elettronici (microprocessori, memorie, ecc.) La garanzia dei prodotti si intende F.co ns. sede, le eventuali spese di trasporto sono a carico del cliente salvo accordi diversi. Per dar corso alla riparazione/sostituzione in garanzia è necessario seguire l'apposita procedura di RMA.

PRIVACY

Ai sensi del Decr. Lgs. 196/2003 la informiamo che i dati trasmessi verranno impiegati coi principali scopi di indagini di mercato e nelle modalità previste dallo stesso, prevalentemente con mezzi informatici. Il conferimento, di norma facoltativo, è obbligatorio per permettere il rapporto commerciale. È in ogni caso fatto diritto dell'interessato esercitare i propri diritti, nei modi previsti dal "Titolo II art. 7" della legge sopra citata, scrivendo a Inware Edizioni srl, Via Giotto 7 – 20032 Cormano o tramite email all'indirizzo info@inwaredizioni.it

---

DIVENTARE MEMBRO INWARE EDIZIONI CONVIENE!

GENIUS MEMBERSHIP: • 11 nuovi numeri di Fare Elettronica in PDF (in coda all'eventuale abbonamento esistente) • 11 nuovi numeri della rivista digitale Firmware (in coda all'eventuale abbonamento esistente), tutto l'archivio delle riviste Firmware pubblicate mensilmente a partire da febbraio 2010 **+ GENIUS Bonus Pack**, ovvero: • accesso al CLUB della rivista Firmware con centinaia di articoli in PDF • accesso al CLUB della rivista FareElettronica (centinaia di articoli in PDF) • due ebook a scelta* Smart card, CPLD, display LCD, PIC, AmpOP, alimentatori, linguaggio C, Basic per PIC, Elettronica Analogica • due raccolte delle annate di Fare Elettronica in PDF (dal 2003 al 2012) e Firmware (dal 2006 al 2012) a scelta.
MAKER MEMBERSHIP: • 11 nuovi numeri di Fare Elettronica in PDF (in coda all'eventuale abbonamento esistente) **+ MAKER Bonus Pack,** ovvero: • accesso al CLUB della rivista Fare Elettronica (centinaia di articoli in PDF) • un ebook a scelta* Smart card, CPLD, display LCD, PIC, AmpOP, alimentatori, linguaggio C, Basic per PIC, Elettronica Analogica • una raccolta delle annate di Fare Elettronica in PDF a scelta* (dal 2003 al 2011).
SMART MEMBERSHIP: • 11 nuovi numeri della rivista digitale Firmware (in coda all'eventuale abbonamento esistente) • tutto l'archivio delle riviste Firmware pubblicate mensilmente a partire dal febbraio 2010 **+ SMART Bonus Pack**, ovvero: • accesso al CLUB della rivista Firmware con centinaia di articoli in PDF • un ebook a scelta* Smart card, CPLD, display LCD, PIC, AmpOP, alimentatori, linguaggio C, Basic per PIC, Elettronica Analogica • una raccolta della annate di Firmware in PDF (dal 2006 al 2012) a scelta

METODI DI PAGAMENTO

Bonifico bancario

Appoggiato su Poste Italiane IBAN: IT 68 I 07601 01600 000070107552 intestato a Inware Edizioni srl.

Bollettino postale

Versamento sul conto corrente postale n. 70107552 intestato a Inware Edizioni srl. Inviare la ricevuta (o copia) del versamento indicando nella causale: "Abbonamento Fare Elettronica"

Carta di Credito

Il pagamento con carta di credito può essere effettuato specificando i dati della carta via posta, email, web, fax o telefono. Per gli ordini effettuati sul sito il pagamento mediante carta di credito verrà effettuato attraverso una connessione sicura SSL per garantire la massima sicurezza.

Assegno bancario

È possibile spedire un assegno bancario insieme a questo coupon.

Paypal

Inviare il pagamento a paypal@invaredizioni.it.